BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 15/173, 15/16 — A1

(11) International Publication Number: WO 00/57296

(43) International Publication Date: 28 September 2000 (28.09.00)

(21) International Application Number: PCT/US00/07577

(22) International Filing Date: 23 March 2000 (23.03.00)

(30) Priority Data:
60/126,017 23 March 1999 (23.03.99) US

(71) Applicant: CORNICE COMMUNICATIONS, INC. [US/US]; Andover Tech Center, Suite 135, One Tech Drive, Andover, MA 01810–2452 (US).

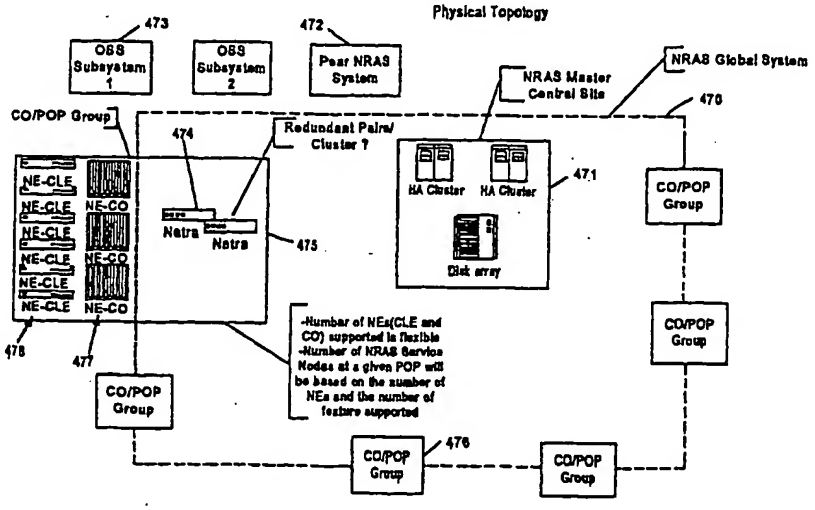
(72) Inventors: LENROW, David, R.; 12 Phinney Road, Lexington, MA 02421 (US). MILLER, Mark, W.; 15 Oak Ridge Drive, Artkinson, NH 03811 (US).

(74) Agents: CORRADO, Thomas, A. et al.; McDermott, Will & Emery, 600 13th Street, N.W., Washington, DC 20005–3096 (US).

(81) Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: A NETWORK RESOURCE ADMINISTRATION SERVER FOR PROVISIONING SERVICES OVER A NETWORK



(57) Abstract

An automatic network resource administration server permits automatic provisioning of network elements based on a user's service request. Rules inheritance and just–in–time expansion enables memory requirements in network elements to be minimized and enables automatic provisioning and administration of huge network. The network resource administration server automatically translates user service requests into a form understandable by one of the specialist or network equipment. The network resource administration contains a core NRAS Manager (170) that includes a core director (60) of all services and multiple Zone Service Managers (50), with each ZSM controlling a physical piece of the network indicated as a particular zone.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

A NETWORK RESOURCE ADMINISTRATION SERVER FOR PROVISIONING SERVICES OVER A NETWORK

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to Provisional Patent Application Serial No. 60/126,017, filed March 23, 1999 by David Lenrow and Mark Miller, the contents of which are hereby incorporated herein by reference in their entirety.

This application is related to Application Serial No. PCT/US99/22773, filed October 1, 1999 and entitled ATM-IP BANDWIDTH ON DEMAND (A-IBOD) by Applicants MARK W. MILLER and DAVID LENROW the contents of which are hereby incorporated by reference in their entirety.

FIELD OF THE INVENTION

The invention relates generally to the field of network communications systems, and, more particularly, to provisioning of services for users over a network..

BACKGROUND OF THE INVENTION

Description of Related Art

Local area networks are well known in the art in which a plurality of subscriber terminals or workstations are interconnected over a network. Typically, local area networks are confined to a collection of devices that are located in reasonably proximity to each other. Wide area networks are similarly known in which stations which are relatively widely separated in geography can be interconnected. In certain network configurations it is desirable and known to interconnect a local area network with a wide area network. Typically, a local area network is connected to a wide area network at a node of the wide area network which is commonly referred to as an edge-switch.

It is commonly the case that large organizations have facilities that are widely separated form each other, such as being located in different cities. It is often required that local area networks for each location be interconnected. Typically such interconnection occurs over a wide area network. It is possible to interconnect a plurality of local area networks over a wide area network in such a manner as to cause it to appear to users of the local area network that they are the only users of the wide area network and that each of the users of the local area network is interconnected with each other user, regardless of location, as if a single network existed linking them all. Such a network arrangement is commonly referred to as a virtual private network (VPN).

Interconnections between local area network interfaces are commonly made using Permanent Virtual Circuits (PVC). Permanent Virtual Circuits are communication paths that are set up in advance and remain established so that they are available for use at any time by a subscriber desiring communications.

A Switched Virtual Circuit (SVC), on the other hand, is selectively established at the beginning of a communications session and torn down at the end of the session. A Permanent Virtual Circuit has the advantage that the signaling overhead associated with establishment and tearing down of a connection between end points is not needed since the PVC is always available. Some types of networks deliver packets of a communications session only on a "best efforts" basis. That means that no special precautions are taken to ensure a given Quality of Service (QOS), Authentication and Authorization, encryption, priority, routing/switching, signaling, compression, address translation, accounting, etc.

Modern wide area networks utilize ATM (Asynchronous Transfer Mode) and frame relay switches and protocols. Other types of switches and protocols are known in the art for wide area networks. A number of protocols are also commonly used for local networks. Increasingly , local area network protocols utilize TCP/IP (Transmission Control Protocols/Internet Protocols) for

communications. This is particularly convenient because TCP/IP is the communications standard for the Internet.

The process of providing service to a user is a complex process involving typically many diverse parties. A user might submit a request for particular telecommunications services to a Local Group Network Manager at the users location. That request might typically go to an Enterprise Network Manager for the enterprise or company for which the user worked. The Enterprise Network Manager might then consider company policies and budget to determine whether the service request should be approved. If approved, the Enterprise Network Manager might then contact a Customer Service Representative of the Enterprise's Retail Communications Service Provider. The Customer Service Representative would take the order for the communications service approved by the enterprise. The a Customer Service Representative might then forward the order to the Billing Department of the Retail Communications Service Provider for credit approval, account set up and billing, and activation of the service requested. Frequently, the Retail Communications Service Provider is a reseller of communications capacity acquired from a Wholesale Provider. Activation of the requested service would then typically involve interaction between the Retail Provider and the Marketing Department of the Wholesale Provider. The Marketing Department of the Wholesale Provider would forward the request for service to a Network Engineer of the Wholesale Provider. The Network Engineer would then actually "provision" the service requested by the original user by setting up the user accounts and network settings needed to provide the service at the computers and other network elements controlling the network over which the requested services are to be provided. In a bad case, provisioning of even a simple service request can take weeks because of the multiple levels of interactions required.

This process is made even more difficult by the fact that the user is usually not conversant with the technical jargon used by the Network Engineer, by the

Marketing People, by the accountants, by the Enterprise and Local Network Managers.

When attempting to interconnect workstations on a local area network over a wide area network, it is highly desirable that the architecture in operation at both the workstation and the underlying local area network remain unchanged from that which it was prior to interconnection with the wide area network. It would also be desirable for applications running on a workstation of a local area network to be able to automatically specify desired features for the connection which is established over the wide area network without any modifications to the application, workstation or local area network and without intervention by a user. It would also be desirable to allow a user of a workstation to specify the features of the connection needed for a particular connection and have those specified features vary from application to application or from session to session to permit an appropriate matching of a user needs with desired features for the communications undertaken. It would also be desirable to gather the necessary information related to a user's connections and to present that information to a provider who provided the communications service.

SUMMARY OF THE INVENTION

The invention solves the problems of the prior art and permits provisioning of service to users of a network to be a simple, fast and reliable process.

In one form, the invention is directed to a network resource administration server comprising a translator for automatically translating user service requests into at least one form understandable by one of a specialist or network equipment.

Another aspect of the invention is directed to a computing device, connected to a network, configured to detect start up of an application and to provide information to a component of said network about the communication needs of that application.

One aspect of the invention is directed computing device, connected to a network, configured to provide information to a user about quality of service options available to the user during a network session.

Another aspect of the invention is directed to a computing device, connected to a network, configured to detect start up of an application and to provide information to a network resource administration server about the communication needs of that application.

Other aspects of the invention are directed to a network resource administration server configured to receive requests for bandwidth allocations to be implemented at a future time, configured to provide billing information about resource consumption by application level processes and configured to provide information about compliance with service level agreements during network use by application level processes.

Another aspect of the invention is directed to a network control configured to represent service requests in terms of logical rather than physical elements.

In another aspect, a network resource administration server is comprised of a rules based translator for automatically translating user service requests into at least one form understandable by one of a specialist or network equipment.

The invention is also directed to methods of provisioning networks by receiving a request for service using terminology appropriate to a user, receiving information from one or more specialists to be combined with information received from said user; and automatically provisioning the network to provide a service requested by said user; and to methods of provisioning a network to provide service, by providing rules for implementing a service to network elements to configure the network element just at the time needed to actually provide the service.

Other aspects of the invention related to techniques for simplifying rules administration in a policy based network, using rules inheritance.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features, and advantages of the system of the present invention will be apparent from the following description in which:

Figure 1 is a block diagram of showing use of is a flow chart showing part of a session initiation process (step 6 of pre-transaction handling). in accordance with the invention in a first network configuration.

Figures 2A and 2B are block diagram showing use of a Network Resource Administration Server (NRAS) in alternative network configurations.

Figure 3 is a block diagram showing NRAS functional components in a network environment in more detail.

Figure 4 a network diagram illustrating the physical topology of hardware in an exemplary installation.

Figure 5 is an alternative representation of the network diagram of figure 4.

Figure 6 is an illustrations of the layers of the Telecommunications Management Network (TMN) model defined by the Telemanagement Forum

Figure 7 is a block showing the software architecture of a Network Resource Administration Server (NRAS) arranged in layers of the TMN model.

Figure 8 is a block diagram illustrating one embodiment, implemented in a workstation, of a Smart Application Environment.

Figure 9 is flow chart of a process mapping a Customer Contact Agent (CCA) User to IP address.

Figure 10 is flow chart of a Smart Application Environment Proxy Signaling Process.

Figure 11 is a flow chart of a Smart Application Environment Marking and Scheduling Process.

Figure 12 is a block diagram illustrating one implementation of a sniffer process.

Figure 13 is a flow chart showing the steps of a sniffer process.

Figure 14 is a flow chart showing destination determination (step 3 of pre-transaction handling).

Figure 15 is a flow chart showing determination of final session information (step 4 of pre-transaction handling).

Figure 16 is block diagram of the Network Service Scheduler (NSS) of figure 7.

Figure 17 is a flow chart of an exemplary process for implementing the Network Service Scheduler (NSS) of figure 7.

Figure 18 is a flow chart showing part of a session initiation process (step 6 of pre-transaction handling).

Figure 19 is a block diagram showing a meta Element Controller (MEC).

Figures 20 and 21 show the contents of Session Start Information (SSI) packets and Session End Information packets, respectively.

Figure 22 is a flow chart showing another part of a session initiation process (step 6 of pre-transaction handling).

Figure 23 is a flow chart used to explain the Just in Time process.

Figure 24 is a diagram illustrating how a service definition is created in accordance with the invention.

Figure 25 is a block diagram showing creation of an exemplary service definition.

Figure 26 is an overview of provisioning using a Network Resource Administration Server (NRAS)

Figure 27 is a diagram illustrating service definition with inheritance.

DESCRIPTION OF THE PREFERRED EMBODIMENT

In policy enabled networks, decisions are made about network resource spending by various devices in order to provide transport performance to serve

applications run interactively by users on workstations. This invention is embodied in most cases in a Network Resource Administration Server (NRAS) that allows those devices to communicate with the originating workstations so that real-time decisions about network resource spending can be implemented at the originating machines. The invention enables enhanced services for billing capabilities such as '900' and '800' and calling cards are created within the same framework.

Emerging network technologies include the ability to control the level of network performance that is guaranteed for a particular application. Administration including circuit management, economic or security authorization, and user interaction are required when WAN costs are about to be incurred or multiple levels of network performance are available. This invention allows a process running on a workstation class of computer to implement programmable administration and real-time logic at the request of network devices.

In one specific application, discussed more hereinafter, this invention allows a WAN switching device to initiate a dialog on a user's workstation in response to his running a videoconferencing application. This dialog application asks the user for various video performance parameters and provides cost information and sample clips for choices called good, better and best. In this application, the video-conferencing application has no knowledge of the WAN switch, yet the user was still presented with choices in real-time.

Some of the novel features of Network Resource Administration Server are:

It enables pre-transport graphical end-user interactivity (See the Customer Contact Agent (CCA) below) with no changes to the network transport or to the application on the host workstation.

Enables the detection of the application type and then the control of the transport. The detection portion can be done inside the workstation, in the transport devices, or inside our NRAS entities that are inside the cloud.

It provides a foundation for implementation of services such as 900 number services for webtone (e.g. 900.sportsclips.com requires pay by the minute QOS guaranteed streaming web video), 800 number services that provide expedited surfing and/or streaming and interactive video with transport QOS paid by the content provider, Prepaid "calling cards" for expedited QOS from any enabled end-station and Data calling cards for using QOS enhanced application services from "foreign" networks such as a customer's LAN or a video-conferencing station in United Airlines' Red Carpet Club.

QoS management including concepts like reservations, bandwidth management, etc.

Provides a platform from which unmodified multimedia applications can access enhanced data services implemented inside Network Service Provider demarc/cloud such as video mail and Local Number Portability/roaming/follow me services based on application detection and CCA.

These features enable significant advantages in Network Administration. These include:

Making policy implementation progressive starting with static rules about network performance needs based on application and user, and then adding varying degrees of interactive user decision.

Enabling 'just in time' and 'tiling' techniques.

Providing pre-transport, real time (human or programmed) decision-making about cost performance tradeoff, and how network transaction will be billed/paid.

Providing final "confirmer" confirmation to make contractual implications of running network applications explicit with expedited performance. This business level acceptance is key/unique.

Provides call detail records to rating/billing infrastructure in real-time.

Provides billing and SLA information to enterprise in real-time.

Provides billing and SLA information to end-user in real-time.

Multi-vendor, heterogeneous technologies supported.

1.1 NRAS Architecture

1.1.1. Figure 1 – Network Resource Administration Server (NRAS)

Figures 1A, 1B and 1C show different exemplary network arrangements in which the NRAS server is conveniently used.

In the arrangement of Figure 1A, the NRAS server is connected to a network that services a plurality of users, including conveniently persons involved in the process of provisioning services such as Users, Local Group Network Manager, Enterprise Network Manager, a Customer Service Representative, Billing Department Representative of the Retail Communications Service Provider, a Marketing Department Representative of the Wholesale Provider and a Network Engineer of the Wholesale Provider

In the arrangement shown in Figure 1B, the NRAS server administers three separate portions of the network as respective zones.

In the arrangement shown in Figure 1C, two NRAS servers administer resources of two different Networks, but are linked to enable provisioning and communications for service requests.

1.1.2. Figure 2 – NRAS Partitioning of Functionality

In Figure 2, the detail shown in figure 1B has been expanded to show how the core NRAS Manager 170 includes a core director 60 of all services and multiple Zone Service Managers 50, with each ZSM controlling a physical piece of the network indicated as Zone 1 and Zone 2 in this figure.

Note that 162 doesn't show as part of either zone and indicates that some equipment at the very core of the network **may or may not** be controlled by the NRAS.

1.1.3. Figure 4 – Physical Topology

Figure 4 begins to show how the actual hardware (Sun Netras or regular enterprise class servers) end up physically in the network. Block 470 is the overall network under management. 471 represents a SB central site component (e.g. where the core manager 60 would run). Block 475 represents one of a service providers many central office (CO) or Point of Presence (POP) sites where the network equipment is at. Within each CO/POP would be one or more of the Netra servers 474 which run the ZSM 50 software. Blocks 477 represents the large network boxes that would typically be physically located at the CO/POP and block 478 which are the enterprise end boxes that are physically at the enterprise sites, but connect into 477 and are physically part of that portion of the network controlled by that particular CO/POP. Block 473 represent the Operations Support Systems/Business Support Systems (OSS/BSS) that are involved in this network (e.g. billing) and block 472 represents a peer SB in another service providers network, which the SB 470 in this network could work with to coordinate services that go between the two network service providers' networks.

1.1.4. Figure 5 – NRAS components in cloud

This figure is a different way to show the same things as Figure 4. Block 440 is the central site controller (a match to block 60 in figure 2 and block 471 in figure 4). Each Central Office/Point of Presence (CO/POP) has a distributed Broker 441 (which is a Zone Service Manager (ZSM) 50 in figure 2, run in a CO/POP 475 on a Sun Netra such as 474 in Figure 4) which communicates to the central site 440 via IP control paths 449. The lines of control 443 show how the ZSM is in control of a series of items: the boxes labeled 442 are the larger density boxes inside the CO/POP (like 477 in Figure 4); the boxes labeled 444 are at the enterprise sites 446 (like 478 in Figure 4); the CCA would live in the workstations 445; the workstations 445 are also where the actual network traffic originates (and the end user applications run.)

Box 447 isn't marked as 'SE' indicating it is a relatively 'dumb' box that doesn't have any of the capabilities needed for service delivery. The NRAS system able to deal gracefully with missing capabilities such as this. It may be able to push that function to a surrogate (e.g. out to an SAE in the enterprise workstations, or back into the CO box 442, or as a software function inside the MEC. The MEC alternative is part of the ESC described below.)

1.2. NRAS Roles/Stages

Figure 7 and the explanation below are both based on the Telecommunications Management Network (TMN) model defined by the Telemanagement Forum, an industry group. As shown in figure 6. The TMN describes the entire data networking Service Provider (SP) Business Support Systems (BSS) and Operations Support Systems (OSS) infrastructure in terms of a five layer hierarchy. From the top layer (broadest, most generic) is the Business Management Layer (BML), Service Management Layer (SML), Network Management Layer (NML), Element Management Layer (EML) and finally at the lowest and most specific layer are the actual network elements ('boxes') themselves. The implementation described is not going to only apply in a TMN structure, but the nomenclature is helpful for the discussion.

The invention described herein includes users interfaces typically considered to be SML entities and involves new functions at the NML layer down through both virtual and physical network elements.

1.2.1. Provisioning (Step #0)

The user interfaces direct information to the Business Management Layer (BML), which in turn interfaces to the Service Management Layer (SML),.

An example of a provisioning request would be the enabling of an application service (e.g. H.323 conferencing) for a given customer. This SML nature request would then be translated, preferably using a rules based mechanism, into a more specific series of actions by the Network Provisioning block in the Network Management Layer (NML). This block would understand the implications on the overall network (that is the role for this layer) and create the necessary changes in a master database and then send the series of lower level commands to all of the Element Management Layer (EML) entities involved. The EML entities would then translate the generic requirements sent down from the NML into the specific commands on the specific protocols that the network elements require. For instance, an access Asynchronous Transfer Mode (ATM) switch at the Central Office (CO) for the first customer node might need to have switched virtual service enabled to allow the H.323 dynamic sessions creation and this ATM switch might be configurable via the Simple Network Management Protocol (SNMP).

1.2.2. Pre-transaction (Steps 1-6)

Pre-transaction refers to the series of actions that takes place before the data (or transaction) part actually occurs.

- Step 1 – User invokes the application. The user invokes the application in one of two ways: through the normal mechanisms on the workstation or the Cornice Smart Application Environment ("SAE" - See Figure 4 above) and Section 1.1). If through the normal mechanisms, then the only way that a new session can be detected for special handling is the Session Sniffer,described below. When the Session Sniffer detects a 'special' session (i.e. one that will be handled in a special manner) it will send this information ahead to the Locator Services module. If the application is launched through the Smart Application Environment, then the SAE will gather up the necessary information about the session (application) being run and send this information ahead to the Locator Services module.

- Step 2 – New session event. Through one of the two mechanisms above, a new session event is generated. This is the first event in the series, signifying that further actions may be necessary to establish a new session connection. It contains the information gathered in Step 1 into a Session Startup Information (SSI) packet which will propagate through the subsequent steps below.

- Step 3 – Destination Determination. Several actions have to take place at this step. Basic role is to determine if the option of special handling exists, based on the termination endpoint requested by the application.

- Step 4 – Final Session Information.. This step determines the remaining details about the setup of the new service, specifically determining what the real set of Service Quality Of Service (SQOS) options should be, rather than just a 'single' 'ideal' one, based on network configuration, current network status/loading, and user preferences.

  The final session setup selection information is added to the Session Startup Information (SSI) packet and the packet is sent along to Network Services Scheduler (NSS).

- Step 5 – New Session Request. This fully formed Session Startup Information (SSI) packet is then sent from the Authorization Communications Controller (ACC) module, through the Network Services Scheduler (NSS) (this allows the NSS to update its info about

outstanding traffic contracts) to the Network Session Control (NSC), in the Network Provisioning module in the Network Management Layer (NML).

- Step 6 – Session Initiation. Based on all the information in the Session Startup Information (SSI) packet from the Network Services Scheduler (NSS), first the peer Network Session Control (NSC), (i.e. the NSC in the zone manager at the destination end) is notified of the new service initiation. Next, the NSC will coordinate the establishment of a path for this session across the network that will perform according to the parameters provided.

  To do this, the Network Session Control (NSC), will instruct the Element Session Control (ESC) to create the service. The ESC will then instruct all of the individual network elements involved in the service in this zone.

  Once the Network Session Control (NSC), in this zone has received notice from its the Element Session Control (ESC)that the local setup is complete, it notifies the peer NSC that it is ready. When it has received notice from all the peer NSCs that they are ready, then the user can be notified that service setup is complete.

  How this is done will be described hereinafter.

1.2.3. Transaction (Steps 7-8)

- Step 7 – Main Data Transfer. All of the data flow for a given session (whether special or default) occurs in this step. This step is primarily the responsibility of the physical network elements. The only unique thing to note is that if a special session was initiated by the Session Sniffer ("SS"), the SS will continue to monitor the control channel data flow to detect an end of session condition.

- Step 8 – Session Teardown. Once an end of session is detected for a special flow (either by the user 'ending' the application in the Smart Application Environment (SAE) or as detected by the Session Sniffer ("SS"),) all of the setup associated with the special flow has to be removed. Specifics will again depend on the nature of the network elements used in the particular network. The data gathered from this special session are collected into the Session End Information (SEI) packet, sent back up to the network layer (NSC module) for post transaction handling.

1.2.4. Post Transaction (Step 9)

- Step 9 – Session Closeout, Figure 21. The key role for this step is to get the session information contained in the Session End Information (SEI) packet into all required system. This involves passing all billing (usage and Service Level Agreement (SLA) information up to the service layer Rating and Discounting module, the Enterprise billing subsystem, and to the user.

  If the Session Startup Information (SSI) packet/ Session End Information (SEI) packet indicates Customer Contact Agent (CCA) end of session billing reporting should be performed, then the CCA is sent the appropriate info. Finally, a short term storage of these SSI/SEI records is performed, ending the life cycle for a service session.

1.2.5. Routine Traffic (partial Step 8 and 9)

Routine traffic (i.e. traffic that will be routed onto default connections rather than getting special handling and incremental costs) is handled in exactly the same way except: the user interaction step may be skipped; the billing interaction may be captured as part of the overall usage of the default channel rather than as a discrete session. For example, Step 8 and Step 9 for the default channel may be done on a periodic basis.

1.3. NRAS Components

The sections below describe the components in Figure 8 in greater detail where the section above has focussed more on the overall system functions.

1.3.1. Provisioning and Configuration Communications ("PCC")

The PCC 14 has two key roles. First, it is responsible for handling the top level provisioning requests for all involved network elements when a service configuration change comes down from the Service Management Layer. Second, it is consulted for authorization any time a dynamic change request takes place during special session control. Part of this consultation may involve coordination with the master traffic engineering module (described more hereinafter to determine feasibility of the requirements for this new special session, or adjustments elsewhere in the network required because of the new session.

1.3.2. Smart Application Environment ("SAE")

This "smart application environment" 23 refers to software technology residing on a computer workstation or other end user network endpoint, which provides application and user context for packet processing without

requiring changes to the applications software. In general, its role is to provide 'extra' capabilities from the originating users workstation that can augment or enhance the overall system operation.

In addition to its core functions, there are four basic functional modules that can be populated within the SAE functionality:

1. Session Sniffing. Associating instances of applications with multiple, possibly dynamically assigned, network traffic microflows. This is done by snooping some or all network traffic and understanding dynamic connection negotiating protocols for various applications. (This is simply a workstation resident instance of the SS, see Section 2.8.3.)

2. Proxy signaling to configure end-to-end dynamic QoS and packet processing for applications without explicit signaling support. (e.g. ATM UNI, IP RSVP, LDP, etc). In this role, the workstation is an active network element and would be controlled as a Virtual Service Element (vSE) by the Element Session Control (ESC).

3. Marking of microflow IP headers. (e.g. Diff Serve Code Point (DSCP) marking) and scheduling of packets from endpoint to network. In this role, the workstation is an active network element and would be controlled as a Virtual Service Element (vSE) by the Element Session Control (ESC).

The Smart Application Environment (SAE) could be implemented via a wide range of application specific ways that would typically run on an originating workstation or thin client. Below are a few examples:

1. Application (Signaling Aware) Launcher: This entity would control starting up an application and communicating its data requirements to a network entity such as an ATM switch, RSVP enabled router, or QVM box by Cornice Communications of Andover, Massachusetts, in advance of starting execution of the application. This would allow an application that was implicit in its data requirements to still 'signal' its requirements without modification of the underlying application. This approach involves creating a wrapper application that has two functions. The first is to start running an instance of the "wrapped" application – a separate standalone executable. The second is to communicate the network requirements of that application to networking equipment downstream. In order to do this the wrapper

application would use networking functionality provided by the host operating system. Messages using ATM signaling, RSVP signaling, or proprietary scheme over UDP/TCP/ATM/FR would notify equipment downstream that subsequent packets with specific classification criteria require specific network performance. This approach would require an application level API for resolving application (e.g. process id) to microflow (e.g. socket number) mappings.

2. Operating System resident application sniffer. This software entity is inserted into the network protocol stack on a host computer system. It sits below the Layer 2 or 3 protocol stack module and above the layer 1 device driver and examines packets being transmitted and received. In this role, the sniffer is programmed to simply forward most packets, but to invoke special logic for packets associated with designated microflows. The logic invoked consists of application specific programmable software plugin modules that understand the specific protocols associated with individual applications such as Netmeeting (H.323 protocol), Oracle, SAP, Citrix, etc. These plugin modules interpret control-plane communications involved in setting up application associated micro-flows. A microflow is one of data channels that might be associated with a particular application session. For example, an H.323 session might require one or more audio, video. and data microflows to service the session. The sniffer entity thus contains complete knowledge of the mappings between instances of applications run by users, and new microflows occurring in the network. The sniffer logic communicates with a signaling module that can then use standard or proprietary communications to deliver both the application to microflow mapping and the application performance requirements to downstream network equipment. The combination of application specific sniffing logic within the operating system protocol stack and an entity to signal on behalf of signaling-ignorant applications yields a completely transparent, end-station based smart application handling environment

3. An intelligent browser plug-in could intercept (snoop) the user actions and send advance 'signaling' information ahead to a network entity so that 'unsignaled' browser based actions could be used unmodified. This approach works well in the case where applications are adapted to network usage by migrating to a web browser based implementation. The browser plugin architectures of most widely used web browsers provide an environment that would allow development of an interposer layer where snooping and proxy signaling could be implemented. This

approach would require application level APIs for obtaining OS (e.g. process id) to microflow (e.g. socket #) mappings

A detailed description of the operating system resident functionality is provided as an example. Figure 8 (orig. 6) is a block diagram of this variant of the SAE.

Proxy Signaling Module

This module can be directed to setup new virtual connections across a connection oriented transport network in support of network bandwidth and behavior requirements dictated by the NRAS Server/ESC. To support actions that require changes to network virtual provisioning, this module uses information provided by NRAS to conduct a protocol/media dependent interaction with downstream network equipment. Information provided by NRAS includes packet classification information and corresponding packet handling requirements.

Likely applications of proxy signaling would include:

- Unsignalled IP to smart IP/ATM overlay using QoS differentiated ATM SVC
- Unsignalled IP to signaling capable IP equipment using RSVP
- Unsignalled IP to Diffserv marking and scheduling module
- RSVP signalled IP to ATM SVC tunneling module

Figure 10 provides an example of proxy signaling for an SAE implementation. NRAS makes a request to the proxy signaling module that includes network requirement specifications and packet classification specifications (510). The proxy signaling module then uses a particular signaling protocol native to the network environment to "ask the network" for needed resources (511). The network can refuse the request (512) leading to an error informing the requestor that the request cannot be accepted (513). If the network approves the request, the proxy signaling module returns to NRAS a success code and information about the forwarding behavior needed (VC assignment, DS marking, etc.) to use the approved (and possibly newly provisioned) resources (514).

Application Snooping Interposer Module

This module represents a workstation resident variation on the Session Sniffer detailed elsewhere in this description.

Flow marking and packet scheduling module

This module would most often be used in a static QoS environment where smart utilization of pre-existing network paths is favored over dynamic creation of new network paths. The policy consumer and interpreter module directs the marking and scheduling module to process packets based on policy and existing provisioning knowledge. Once the policy module has determined a marking syntax (e.g. Diffserv bit settings) that has network –wide meaning, the marking Process involves simply labeling all packets associated with the flow before forwarding. In addition to this marking, which determines the behavior of packets elsewhere in the network, it may be required that the local forwarding interface(s) prioritize and schedule traffic among multiple dissimilar flows so as to enforce locally any end-to-end policy objectives. This typically involves selecting one of potentially many pre-existing queues with differentiated scheduling behaviors.

Figure 11 provides an example of an SAE marking and scheduling module. This module uses some packet classification criteria supplied by the calling module to determine whether any special marking (departure from default mark) is required (520). For packets retaining the default mark the forwarding queue mapping is looked up (523) and the packet enqueued (524). For packets classified as requiring special handling, the packet header marking for the classification is applied (521) and the forwarding queue mapping for that class looked up (522). The packet is then inserted into the proper queue (524). All queued packets are scheduled based on a variety of widely used packet scheduling algorithms such as WFQ, CBQ, etc. (525).

1.3.3. Session Sniffer ("SS")

The Session Sniffer represents an entity that checks every IP packet that travels 'through it' (i.e. it has to be in the data path) to determine if the packet is part of an application that requires enhanced network handling. The sniffer has to recognize the beginning of a 'new session' so that other modules in the system can determine if they have incremental work to perform, and 'end of session' so that the system can remove the service and perform end of service billing.

This module examines the payload of specific IP microflows and has enough context to correctly interpret the control communications that are encoded in the packet exchanges between end-stations. In order to provide application snooping support of this type for a new application, it is necessary to understand the network communications used by that application to negotiate dynamic network routing information. This typically results in a preconfiguration of a sniffer for application X. This can then be associated with plugin logic **a** with any packets addressed to UDP port 10000, for example. The logic in plugin **a** would typically be able to understand protocol **p** for application X. Protocol **p** might negotiate communication using protocol **q** on UDP port 15000. At this point the sniffer might report to the policy system that there is an instance of application **x** starting up between source address **s** and destination address **d** using UDP port 15000.

The session sniffer component can be implemented in a variety of different locations:

The SEE-1000 by Cornice Communications implements a rules engine as part of its per packet handling to perform the session sniffing role.

A session sniffer could also be implemented on the endpoint workstations as a 'shim layer' interposed between the application and IP stack. On Windows NT, such a layer would be in the data plane for every packet sent and received and would perform this sniffing function.

A key third way is as a component of the NRAS server directly. Such an approach is preferred in many network environments and is shown more in detail in Figure 12 (orig 13). In this configuration, the sniffer is part of the NRAS manager, block 170. As such, it would be part of the distributed architecture, where multiple instances of the manager would run, with one instance in each zone (the Zone Service Manager 'ZSM') block 171. Within the ZSM, many instances of the actual session sniffer, block 200 would exist, probably one per enterprise site in that zone.

Figure 13 (orig 14) shows a flow chart of the process.

To operate, the SZM would create a new session sniffer, then create a data connection 172 from one of the network elements involved in the control data path for that service (e.g. block 161) into a driver 201 of the proper type (i.e. 201 is an ATM driver if 161 is an ATM network element.) The session sniffer receiver 202 would then register to receive any packets that arrive at the driver 201 for this sessions control data path.

In block 210 the driver receives the packet from 161 and then sends it to the receiver in block 211. The receiver (which is aware of the low level details about the type) in block 211 then would format the packet into a generic/internal format for further processing by block 203. Block 203 would do internal lookups necessary to match to packet to the specific service session taking place in block 213. The next stage is then to match the L3/L4 in block 204, which identifies the actual application in block 214. The specific application type specific detector block 206 is then called.

If upon execution of block 215 by the application detector, no application event (e.g. start of a new flow within the application or application complete) then logic block 216 'yes' path is followed without further application specific handling. If an application event is detected, i.e. following path 'no' from logic block 216, the block 217 will send the flow event to the controlling ESC for this service.

If the L3/L4 flow is marked for packet return (the 'yes' path to logic block 218), then block 204 will send the packet along to block 205. In block 220, the packet will be formatted appropriate to the type, so that it can be sent back out in block 221 via 201 to 172 to 161.

1.3.4. Locator Services ("LS")

The Locator Services (LS) module, block 22 of Figure 3, is an important module in determining the destination for a new service flow. See Section 1.2.2 , Step 3.

Figure 14 details the operations of the Locator Services (LS) module. When the Session Startup Information (SSI) packet arrives (230) first the source IP address (which is always known at this stage) is put into the SSI (231) and then next thing that needs to be done is to determine the destination IP address which is done by block 22 the Locator Services ("LS"). The destination can sometimes be directly resolved from the data flow, in which logic block 232 goes immediately on to storing it into the SSI 236, or it may be a logical descriptor of the destination in which case

logic block 232 proceeds to logic block 233 to continue the resolution process. This logic block determines if the destination hostname is specified, e.g. fredspc.flintstones.com, in which case logic block 233 proceeds along the yes path to block 234, which would use DNS for the given service to resolve the IP destination hostname into an IP address and proceed to 236 to store it into the SSI. Lastly, if block 233 indicates that it is not a hostname that's specifies, it must be a user (e.g fred@flintstones.com) and the virtual ILS service would locate that users current IP address and proceed to 236 to store it into the SSI.

1.3.5. Authorization Communications Controller ("ACC")

The Authorization Communications Controller (ACC) module, block 15 of figure 3, is responsible for several things (See Section 1.2.2 , including the later stages of Step 3 and most of Step 4 of section 1.2.2.

First, it takes the NRAS abstracted form of the requested communication and translates it to the Network Layer specific form. Secondly, it performs the actual low-layer communication with the appropriate network layer sub-system.

Further, it is responsible for interactions for the Enterprise Authorization. It takes the NRAS abstracted form of the requested communication and translates it to the Enterprise specific form. Secondly, it performs the actual low-layer communication with the appropriate Enterprise system, e.g. the Light Weight Destination Access Protocol (LDAP) transfer.

Continuing with Figure 14 , once the destination IP address is determined control continues in the ACC block 15. Logic block 237 determines whether this source and destination IP address pair are part of the same VPN. If they are not, then block 238 select a Service QOS (SQOS) from the default set and proceeds to logic block 240 to complete this Step of SSI. If they are part of the same VPN (the 'no' path from block 237, then the SQOS from that VPN definition (which can be one or several) are sent along to block 240.

The new information is added to the SSI packet in block 240 and it continues along in the Authorization Communications Controller (ACC) in block 241.

Going to Figure 15 now, with core service definition from block 250 (information such as services purchased, rate limits imposed, billing options enabled) the Authorization Communications Controller (ACC) module consults the Network Management Layer (NML) (specifically the

Network Services Scheduler, block 13, in the Network Provisioning block) for the range of options available to this customer from the source location to the destination location. (This consulting may result in constraints such as advance reservation requests, static provisioning and current network loading.) See Figures below for details on the NSS.

First, logic block 252 checks to see if the NSS refused any special session, then the session is marked to take place on the default session block 261.

Otherwise (the 'yes' path from 252), then the response is checked to see if the equipment path is capable of the requested special in the VPN definition. If it is not, then the 'no' path out of block 253 will proceed and the session is marked to take place on the default session block 261.

Otherwise (the 'yes' path from 253), then the response is checked to see if the equipment path has the capacity to support the requested special in the VPN definition. If it is not, then the 'no' path out of block 254 will proceed and the session is marked to take place on the default session block 261.

Otherwise, (the 'yes' path from 254) the ACC next gets the authorization information from the enterprise for this particular user. If this user is not authorized for special session creation, then the session will take place on the default session block 261.

Otherwise, the constrained list of options remaining (and especially the billing options, e.g. credit card or calling card) is put into the SSI in block 256. If there are not multiple options remaining (that need to be resolved by the user) and the Service Definition is not marked to force the CCA presentation, then block 257 will follow the 'no' path to block 259. Otherwise the 'yes' path from 257 is followed, resulting in user interaction via the CCA for final determination. If the user chooses to use no special session, the session will take place on the default session.

At this stage, the SSI is checked to see if the VPN has be selected for default service, which would follow the 'yes' path to block 261 for default setup. If not, it would follow the 'no' path, in which case the final special setup related to this special service in block 260.

1.3.6. Network Services Scheduler ("NSS")

The Network Services Scheduler (NSS) is an important module in the overall NRAS approach. It is described in figures 16 and 17 . It acts

as a central coordinator for resource needs of network traffic and resource capabilities of the network. When another module (NSC, ACC) requests resource availability information for network traffic, the NSS first checks whether currently provisioned resources have capacity available to support the requests. This decision requires information about existing usage by live traffic and information about reservation commitments made that could be redeemed during the period under consideration. Specifically, this means the Existing Resource Usage Manager (ERUM 130) uses the Aggregate Path Admissions Accounting (APAA 133) to determine how much of various required resources are available in currently provisioned paths. The ERUM, then uses the Reservation Manager Database Client (FMDC 132) to learn about any reservations claiming additional resources. If there is sufficient available resources based on the combination of usage and reservation data, the ERUM is able to return status to it's client indicating that resource preferences can be met. In the absence of sufficient provisioned resources, the New Resource Allocation Manager (NRAM 131) can attempt to dynamically allocate additional resources by contacting the PCC 14 of figure 8.

The proxy signaling capabilities, an example of which is described in the section above on SAE, provide a system in which any application requiring specific network behavior can use a control mechanism to indirectly ask for it. This fundamentally changes the nature of communications from unregulated and unpredictable resource utilization to where it is possible to impose tight management. The Network Services Scheduler (NSS) can augment any existing network resource management intelligence such as that contained in a UNI ATM switch. When ATM signaling requests a new Switched Virtual Circuit (SVC) from the switch, the NSS is not required to determine whether the switch has sufficient buffers and other resources to allow the request. In the case of an environment where multiple IP microflows are multiplexed onto an aggregated ATM permanent virtual circuit, the VSS performs the role of managing access to the network resources associated with that PVC. Applications in such a QoS architecture can ask the NSS to admit them to an aggregate forwarding path. This requires that the NSS keep state regarding all admitted traffic flows and the resources left available given commitments to the admitted flows. In addition to resources that may be committed to other active applications at any given instant, reservations previously accepted by the reservation system must be accounted for. The NSS must insure that when the reservation holder asks to be admitted, sufficient resources are available.

The notion that performance sensitive applications can be caused to describe their specific requirements does not imply that it is reasonable to have all applications behave this way. Many applications whose data traverse the network have little or no performance sensitivity and are quite well suited to simple best-effort delivery. Other applications needs can be met by a range of course grained classes of service. Still others need fine grained, guaranteed quality of service. A side effect of the fact that networks must carry diverse traffic types is the ability to exploit the complementary nature of these different types. If all network traffic relies on hard guarantees and committed resources, incremental services can only be added by provisioning additional resources. In a network with mixed best effort and guaranteed traffic, additional guaranteed traffic can be admitted by degrading the performance of the best effort traffic. Similarly when there is low utilization by guaranteed and reservation based traffic, the unused resources become available to the best-effort classes providing an enhanced best effort experience. Cornice refers to resource management in a mixed traffic environment as exploiting various "pools of bandwidth".

More detailed discussion of the details of implementing a reservation system within a given pool of network resources can be found in the co-pending application describe above.

An example of a software overlay implementing admission to multiple traffic classes based on pre-provisioned ATM PVC's demonstrates the ability to offer a diverse, dynamic set of services exploiting the complementary nature of guaranteed and best effort resource pools. In this example two ATM PVC's have been pre-provisioned between two enterprise sites. The slowest link between these two sites is a 1.5 Mbs T1 link. One PVC is provisioned as variable bit rate, with a guaranteed minimum capacity of 1.0 Mbs. The second is provisioned as available bit rate and with no performance guarantees. An intelligent edge router can forward "don't-care" flows to the ABR path, and "special" application flows to the VBR. If special traffic admitted is zero, the best effort service gets the whole 1.5 Mbs, but up to 1.0 Mbs of "special" flows can be admitted as incremental high quality, with the acceptable side effect of degrading the best effort service. Subscribers of a best-effort service with loose or nonexistent guaranteed have effectively agreed to have their service degraded at the convenience of the service provider. By selling a mix of best effort and premium from a single physical network, the premium services are essentially subsidized by the infrastructure required to provide the best effort service.

The Network Services Scheduler (NSS) acts as an overlaid call admission and control (CAC) server. In an NRAS network, system flows can only be admitted to non–default traffic groups by the consent of the NSS module. This module provides an "outboard resource accounting function" to build network resource utilization knowledge as an enhancement to shared or aggregated forwarding paths. Native networking accounting (e.g. ATM SVC) may still be used for admitting or denying requests for new aggregate circuits (made by the Provisioning and Configuration Communications (PCC) module), but admission to any of the resulting circuits is at the discretion of the NSS.

The following discussion of the flow-chart of figure 17, the narration is based on an aggregation model in which the special applications (those getting better than best-effort forwarding) supported by the premium aggregate are Voice over Internet Protocol (VOIP) and videoconferencing. VOIP is counted in units of admitted phone calls, and videoconferencing is counted in units of bandwidth consumed.

In the description of block #300, logic in the NSS becomes active when a client module (NSC,ACC) requests admission of a new application flow, which policy interpretation has determined should get special handling. Resources for outstanding reservation commitments are always considered before any admission decisions so, by definition, if the user, application, time, etc. indicate that this is a flow for which a reservation exists (determined by query to reservation module in 301) there are adequate resources available and the flow should be admitted (#302) if the user presents the correct reservation ID. If the new flow did not have a reservation is it tested for association with the VOIP application (303). For VOIP flows, the aggregate path admissions module (133) would test whether the allowed number of supported phone calls are active across the circuit between the communicating parties (304). If the limit has not been reached, the new flow is admitted, and the count of current VOIP calls is incremented (306). If the maximum has been reached, the new resource allocation manager (131) would ask the PCC to allocate more VOIP capable circuit capacity between communicating parties (305). If new resources can be allocated, the application test logic (starting at 303) is re-entered with the expectation that the sufficient resource test (304) will now succeed. If new allocation failed admission is refused with an error code describing lack of resources. For flows not associated with VOIP, the first application match test (303) fails, and the flow is then tested for association with the video-conferencing application (308). If this test fails, the logic which lead to a special admission request has failed and an error

is returned indicating that there is no special resource allocation policy for this flow (311). If the flow is associated with the videoconferencing application, the bandwidth requested for the flow is compared with the available bandwidth for the existing circuit(s) between the communicating parties(309). If sufficient bandwidth is available, the flow is admitted and the available bandwidth decremented by the amount of the request (312). If not, the PCC is asked to attempt to allocate new Videoconferencing bandwidth between communicating parties (310). If the PCC is able to allocate new resources, logic resumes at the videoconferencing application test (308), with the expectation that the resource availability test will now pass. If the PCC is unable to allocate new resources, admission is refused with an error code indicating lack of resources (313).

1.3.7. Customer Contact Agent ("CCA")

The CCA handles all user interactions. These may be in the form of service provisioning (most likely Web browser based) or per session specifics gathering (where the Browser interface style may or may not be appropriate). Examples of the information that might need to be collected are: price vs. performance trade-offs in a session setup (see Figure 23 (orig. fig.10)), action desired on a 'busy signal' or 'resource unavailable', entry and confirmation of a calling card number or password, billing and SLA information at the end of session. Some of the interface options include:

- Standard browser based interfaces, either HTML or Java
- Standalone Java based interfaces
- Standard Windows interfaces on Windows 95/98/NT.
- X Client interfaces on a wide range of end station types.

The CCA also provides NRAS with User identification, i.e. associating users with IP addresses. This requires an ability to know about user login events (to either the operating system, or an adjunct authentication server) on the end-station it is running on. This User to IP Address Mapping module can exist in either user space or kernel space. It must obtain awareness of user authentication events in order to associate a user entity with the IP source address of its end-stations. This module communicate this mappings to NRAS as soon as they are known. In a preferred implementation, the User to IP Address Mapping module is a user mode Windows NT process, run with administrative privileges, that discovers and caches the Windows user-name and domain of a workstation and provides IP address to Windows user mapping services to NRAS. For instance, for Windows NT workstations, a process can

obtain user info by query to the Windows registry, a database that is real-time populated by the Windows user authentication tools.

More details on the CCA can be found in the co-pending application, referred to above.

1.3.8. Network Session Control ("NSC")

This module in central to the NRAS architecture. It is responsible for directing and synchronizing all per session setup and teardown across the entire network. It does so primarily by interacting with the ESC below.

Figure 19 details Step 6 of figure 3 which is mostly activity by the Network Session Control (NSC). Based on all the information in the SSI packet from the NSS (block 270), first the peer NSCs (i.e. the NSC in the zone manager at the destination end) is notified of the new service initiation in 271. Next, the NSC will coordinate the establishment of a path for this session across the network that will perform according to the parameters provided.

To do this, the NSC will instruct the Element Session Control (ESC) to create the service. The ESC will then instruct all of the individual network elements involved in the service in this zone, blocks 272-274.

Once the NSC in this zone has received notice from its ESC that the local setup is complete, it notifies the peer NSC that it is ready in block 275. When it has received notice from all the peer NSCs that they are ready, then block 277 can notify the user that service setup is complete.

1.3.9. Element Session Control ("ESC")

This module, under the direction of the NSC, is responsible for circuit setup (which may be very different on different topologies and equipment) and for circuit teardown when the session is complete for all elements under its control. Specific examples depend largely on transport types:

- In the Cornice QVM, referred to above, ATM UNI Signaling is performed by the STC to create VCs with the exact QoS requested.
- In the NRAS, each Element Session Control (ESC) would use a Third Party Call Control (TPCC) such as Soft PVCs under ATM UNI standards to create/destroy the VCs.

- Another configuration would have the ESC participate in the Label Distribution Protocol (LDP) to label and move the session onto the right route in a Multi-Protocol Label Switching (MPLS) based WAN.
- Policy based schemes with ESC acting as Policy repository or policy decision point.
- Another variant would have the Network Session Control (NSC) in a TAPI proxy in a workstation.

As described above and shown in Figure 19 (orig. 23), the NSC commands the ESC when to establish/create a new service. The ESC will then instruct all of the individual network elements involved in the service in this zone, blocks 272-274, with the appropriate set up. The actual actions performed will vary widely, but a couple examples might be:

- In an ATM network, the Customer Premise Equipment (CPE) endpoints would be told to proceed. The endpoints would then initiate standard SVC creation procedures with the QoS guidelines presented in the SSI packet.
- Similarly, in a non-ATM network, the ESC module might change some Layer 3/Layer 4 routing information in the CPE box via SNMP and then initiate a third party call control process on the access box to steer the data across the network to the destination access box.

The destination end would be configured in a similar manner by its controlling ESC.

Figure 20 shows a top level view of the ESC. Two key concepts are involved in the design of the ESC. First, all capabilities are decomposed into primitives called virtual Service Elements ("VSE"), for instance block 502. These in turn actually map onto a mixture of network elements and software entities that are tied together as one 'super' entity referred to as a meta Network Element ("MNE") block 500 controlled by the Meta Element Controller (MEC.) The concept of the MKNE/MEC allows the NRAS Manager to assume logically uniform capabilities in the underlying network elements 505, because either through physical (with separate drivers 503) or logical NEs (like 506) the capabilities necessary to implement the desired services will be implemented by the MNE/MEC.

1.3.10. Transport Services ("TS")

Transport services (TS), shown in the Network Elements Layer of figure 3 is responsible for moving packets through the network according to all of the specifics set up for a given session. This involves two key aspects:

Routing

Once the STC has established all the necessary configuration information, the TS module must deliver all packets onto the appropriate 'circuit' based on the L3/L4 or 'label' information in the packet.

Traffic Management

Further, the TS must make sure the required 'quality' of each session is maintained to the extent possible. For instance, multiple sessions on the same 'circuit' may need to have packets re-ordered so that higher priority traffic is passed first (for instance with a technique like class based queues CQB). This can also involve ensuring that a given 'circuit' doesn't exceed a committed rate, typically called 'traffic shaping.'

1.3.11. Session Monitoring ("SM")

Session monitoring is responsible for tracking packets through the network according to all of the specifics set up for a given session. The transport elements must perform this action, under the control of the ESC, which tracks the application progression with the Session Sniffer. If the transport elements don't perform these functions, then NRAS can still provide billable services, but not ones based of usage or SLA. In general, the session monitor involves two key aspects:

1.3.11.1. Stats

Full accounting on a per flow basis is required, e.g. packets and bytes, receive and transmit.

1.3.11.2. SLA/Performance monitoring

Further, a more proactive role is needed for key performance critical 'circuits' e.g. Operations Administration and Maintenance (OAM). Per VC Performance tracking and raw reporting and reporting of 'violations' can then take place. This module is also responsible for any gross anomalies, e.g. Loss of Signal or AIS events.

1.3.12. Performance and Alarms Collector ("PAC")

Via the ESC, all information is gathered and provided to the PAC module. This module then performs a translating (from the NRAS internal/abstract format to the OSSs specific format) and communicates it back to the OSS. For example, an SNMP Trap could take place to notify of a failure on a given piece of network equipment. The ESC would put this into generic NRAS format and provide to PAC. PAC could then provide it to the OSS in a form specific to the OSS system, but completely independent of the network equipment type. Billing Functions Controller ("BFC")

1.3.13. Billing Functions Controller (BFC)

The Billing Functions Controller (BFC) (item 10 of figure 8 is the coordinator for any billing related activities the NRAS will participate in. For instance, if a session has incurred incremental charges, both the enterprise (for departmental allocations) via the Enterprise Billing Communications module, and the service provider (to bill the enterprise) via the OSS Billing Communications capabilities, would receive some form of billing update. Note: the NRAS has its own internal billing abstractions, Session detail records (SDR) and Call Detail Records (CDRs).

Internal Session Detail Records (SDR) and Call Detail Records (CDR) are created in the Billing Functions Controller (BFC) and translated then sent to the Service layer Rating and Discounting module of the Service Management Layer.. These can be based on standards, e.g. Metratech and Netflow and RADIUS, or proprietary systems. This module is also responsible for more involved transactions across to bordering service providers for settlement, e.g. transfer of payment resulting from 1-900 access (although this may be handled by the Service layer.)

The contents of the Session Start Information (SSI) packet and Session End Information (SSE) packet are shown respectively in figures 21 and 22. The use of these packets will be discussed more hereinafter.

Figure 23 details Step 9 of the NRAS processing. When the Session Start Information (SSI) packets and Session End Information (SSE) packets associated with this service arrive at the BFC, block 285, it needs to first collect the SEIs from its peer BFCs in all other involved zones, block 286. Next, it will extract the billing options lists from the SSI (block 287), so that it can begin the interactions with all participants in this list.

First the SEI info is extracted into the proper format (also specified in the SEI), and sent along to this billing host (block 288). This entry in the billing list is then checked to see if it is marked as the master in block 289. If it is the master (path 'yes'), then the host is contacted for retrieval of the billing amount in block 290. In either case, block 291 then checks to see if there are any remaining entries in the billing list, returning up to block 288 if there are (the 'no' path), or proceeding to block 292 is the 'yes' path is followed.

If the SSI/SEI indicates the enterprise wants billing reports, then logic block 292 'yes' path will be followed, with block 293 sending the report in the format specified, to the host specified. In either case, control proceeds to logic block 294.

If the SSI/SEI indicates CCA end of session billing reporting should be performed, then logic block 294 'yes' path will be followed, with block 295 sending the report to the CCA. In either case, control proceeds to logic block 296.

Block 296 represents a short term storage of these SSI/SEI records on session complete, ending the life cycle for a service session.

1.3.14. Enterprise Billing Communications ("EBC")

Internal SDRs and CDRs are created in the BFG are received by the EBC and translated then sent to the Enterprise billing system. These can be based on standards, e.g. Metratech and Netflow, or proprietary systems.

1.4. System Implementational Concepts

1.4.1. JIT Policy

A key innovation differentiating Applicants' work from conventional policy networking is captured by the term "just in time" policy.

In order to scale policy networking to support the biggest existing and future service provider networks a potentially fatal bottleneck can occur with respect to the number of packet handling rules a given switch or router can support. For a service provider with 10,000 enterprise customers, each employing an average of 1,000 employees, with an average of 10 closed group application VPNs defined, a worst case scenario could require that 100,000,000 rules be resident in a given network element to ensure the ability to completely specify possible network configuration and behavior. Certainly computer storage

devices such as hard disks and DRAM can easily store 100 million rule definitions. The bottleneck for wire speed forwarding equipment with Layer 3/4 hardware optimization is typically the size of the rule logic memory in the forwarding hardware. Currently one estimates that the huge majority of wire-speed IP forwarding rules engines being developed support on the order of 100,000 rules. At no time in the foreseeable future can one anticipate hardware economics allowing these tables to grow the three orders of magnitude required to support really big networks. A number of innovations have been implemented to allow rules to be populated on-demand, in response to actual network usage rather than the brute force approach of anticipating all possible usage scenarios.

In this description, the process of interpreting the abstract logic encoded in a policy rule is referred to as "policy expansion," that is, invocation of a rule may trigger activation of other rules implementing the policy rule invoked. In the implementation described, the rule is specified in terms of abstract entities such as users and applications that must be mapped by some logic onto network entities such as IP addresses and packet flows. In addition, policies may specify aggregate entities (groups of users, wild-card matching for addresses, ports, etc.), allowing efficient policy description, at the expense of requiring logic to implement the aggregate classification. An example is a policy rule stating that all n users from enterprise A should get "platinum QoS" for application X. On some level this rule requires that the enterprise A abstraction be expanded into the specific group of user abstractions that must then be mapped to IP addresses. Furthermore, the application X abstraction requires some, probably dynamic, assignment to specific packet flows. At the wirespeed classification device, this could mean n^2 rules for every possible user to user combination.

In contrast to the enormous number of rules required to statically configure policy networks in anticipation of all possible network events, the "Just In Time" rule expansion implementation of this invention allows sparse population of the actual rules residing in a given classifier. Central to this optimization is the observation that when application events (e.g. begin and end of an instance of an application between address a and address b) can be discovered in the control plane, they can be used to initiate just in time adjustment of the data plane configuration to achieve the desired behavior for active applications, and to remove rules configured for applications that have become inactive. The application sniffer implementations described above, describe a way of detecting application events in the control plane. In general, it will be the case that the latest possible expansion of rule elements leads to the greatest conservation of rule space with respect to active rules.

Another benefit of the late expansion of policy is that it allows an opportunity to create rules that describe multiple possible actions in response to a given condition, explicitly deferring the decision about which one to choose until the event actually occurs. Among the benefits of this capability are:

- Real-time/run-time system state may be included in the decision making process. Policy choices can be shaped by network resource availability at run-time, rather than pre-configured choices. For example, a policy may specify that the QoS delivered for a given user/application should ideally be x, but that anything above y that can be negotiated with the bandwidth broker software module at run-time is acceptable. With a static policy system the creator of the policy is forced to choose between always getting the lower quality (y), or else getting a virtual "busy signal" from the network when resources are available above level y, but not sufficient for level x.

- System user input (e.g. Fred at enterprise A, sitting in front of the keyboard). In many cases the need for specific network behavior is a function of what the user is intending to do with a given application, not just which application he is using. No policy in the world can tell the difference between a videoconference in which Fred and Joe exchange fishing tips, and one in which they are discussing a critical business development. Their quality preferences and price tolerance could be quite different in the two cases. The Just In Time Policy system allows a policy rule specifying an operating envelope to be further discriminated at run-time. For example, a policy might specify that when Fred and Joe collaborate with some videoconferencing application, the initiator should be given a choice of either "High", "Medium", or "Low" quality. When Joe calls Fred, he sees a popup window on his screen asking which of the three choices he prefers for this collaboration session.

This following greatly simplified example shows an attempt to match a single application instance with a single (complex) policy rule. In practice many such rules may exist and all application events must be tested against the full suite of rules, each test of which may look like the following.

As described in Figure 24 an attempt to identify traffic that may require special policy-defined handling begins when an "application begin" event is detected by the control plane application sniffer and sent to the policy expansion logic (400). Such an application event communicates information about the application started and the addresses of the communicating parties (401). First the user mapping is obtained by the CCA for the IP address that initiated the communication (402), then the associated user-id is tested for membership in the group for which the sample policy applies (403). If the user is not a member, and thus not affected by this rule, rule expansion is not done, the rule has not been matched and the application will get whatever the default

behavior (404) for traffic is (e.g. best effort forwarding or drop the packet) If the calling party user was a member of the group, the user mapping for the receiving workstation is queried (405) and tested for group membership (406). If the receiving party user is not a member of the group specified for special handling, the potentially special caller is calling a non-special party and default behavior is the result (407). If both users are members of the group, special treatment is required and specifics must be determined. First the NSS is queried to determine which of the policy specified range of choices are valid to present to the user at this instance (408). Once choices that may be invalid due to instantaneous resource availability are removed from consideration, a graphical dialog box is presented to the user by the Customer Contact Agent CCA, returning the users preferred performance choice (409). At this point, specific rules will be populated in the devices forwarding packets between the two communicating workstations (410). This will typically involve at least one rule per microflow associated with the application. This system stays in this state until the end of the application is detected by the control plane sniffer (412). When the application end event is received (411), the specific rules installed for this application instance are removed again to optimize use of the rule memory in forwarding devices (413).

1.4.2. Service Abstraction

1.4.2.1. Functional Description

NRAS presents the operation of the network to the user in terms and according to attributes and language that correspond to the normal user view of the world. In other words, most users of a networking system are **not** networking experts, but rather experts in their own little corner of the world. For instance, a Customer Service Representative knows almost nothing about the underlying technology of the network they might be receiving order requests for, they might only know that all information on Form 1234 has to be completely entered before a new service can be started, or that a customer may be on credit hold for a particular reason. Similarly, a network engineer has no idea what the business model or market strategy might be for introducing a new service, only that its based on the latest and greatest ATM technology they just finished bringing up. Bottom line, no one person knows everything, and most of the users do **not** understand the technology of the network.

Rule based translation provides the ability for most of the users of a 'service definition' system to define the desired functionality in very high level abstractions that are natural to them and not in techno jargon which

is completely meaningless to them. Other forms of translation, other than rule based, are of course possible.

1.4.2.2. Service Definition

Figure 25 illustrates the power of this translation. Block 450 shows the types of things that most users deal with in their daily lives. They think in terms of business models (e.g how much does this cost) and applications (I use NetMeeting for videoconferencing and Baan for ERP) and on Friday morning at 11:00am EST I am doing a collaborative design review on my PADS printed circuit board layout with Fred in Atlanta and Joe in Chicago. Except for a handful of actual networking experts, most users of a network don't know or care much about it at a technology level, just about whether they can get their jobs done when it happens to involve the network.

These top level abstractions have to ultimately be 'translated' into terms that the actual network equipment can be configured with. With the class of highly dynamic services desired by enterprise users today, this is an activity that would take place thousands or even millions of times per day. In accordance with the invention, this has become a fully automated function. In fact, this is one of the core functions of the NRAS, taking these abstractions and the rules for translating those abstractions into the core network elements in a fully automated fashion.

As noted elsewhere, NRAS also interacts out to the end users (452) and also is aware of services which terminate at a server (e.g. 454) instead of another user.

1.4.2.3. Exemplary Service Definition

Figure 26 shows an instructional view of an exemplary service definition. Block 431 shows how a user view of a VPN is very simple. Tom, Dick, etc. are part of a closed group VPN that does NetMeeting. As the rest of the figure shows, there is a lot that goes on 'behind the scenes', but a user of the network. defining their usage of this service, does not need to know than what they want to do to use the network. This is described more hereinafter.

1.4.3. Specialized Views

Since NRAS presents the operation of the network to the user in terms and according to attributes that correspond to the normal user view of the

world, it can also present the individual pieces to each user separately and then join them together as part of an automated system.

1.4.3.1. NRAS Provisioning

Figure 27 shows how the NRAS has separate User Interface (UI) screens for the many different users (roles) involved in provisioning. For instance Wholesale Service Provider (WSP), Retail Service Provider (RSP) and Enterprise Network Manager (ENM). (One may refer to this slide as the 'hydra' because, as drawn, the many different views (screens) of the provisioning information shown along the top look like many 'heads' on a hydra.) Each user only needs to define that part of the system for which he is the specialist. For instance, as described above, the end user is only a specialist in the field of knowing precisely what they want to do at any given time. A more traditional specialist or 'expert' would be the individual who understands exactly what network behavior is required for a given application to work well, e.g the 'application XYZ expert'. Likewise, a 'network engineering' expert would have to define the specific transport specific attributes necessary to achieve a certain behavior (e.g. if an application expert says a 50 millisecond round trip is needed, the network aspect could define the cell delay parameter for an ATM network.)

The series of User Interfaces (420) splits up the presentation of the problem according to each of these different expert or specialist views. The individual pieces are stored as XML Service Definition Components 422 at the Central Site 421. The Central Broker 423 consolidates all of the Service Definition Components into overall Service Definitions, and then decides on the distribution to each of the Zones, sending the per zone info 424 to each of the Zone Service Managers 425. (See Service Definition Construction below for more detail.)

1.4.4. Service Definition Construction

1.4.4.1. Functional Description

NRAS presents the operation of the network to the user in terms and according to attributes that correspond to the normal user view of the world. This results in a simpler view of the world to these users, but a more complicated view to interpret by the NRAS system. This is a good

tradeoff though, because you have an automated computer process dealing with the complexity instead of a human!

However, it is difficult to scale this solution adequately to serve the need of the very large network service providers. A scheme which will allow a huge number of these service definitions to managed simply is needed. In accordance with the invention, NRAS implements a system where a single master copy of any single component type's attributes are inherited by all users of this component. In this way, if the attributes of a master component are modified, all services based on this component will automatically 'inherit' that change next time they are run.

Considering Figure 28, for instance, lets say in our example above that Joe and Fred are enrolled in Acme Tool Company's Video Conferencing group which is bought from NSP1 (a data networking retail service provider) who in turn buys its networks services from WSP1 (an infrastructure based data networking wholesale service provider) The way the service definitions are constructed is as follows:

- WSP1 sells its video conferencing to NSP1 by defining a standard service offering with 5 levels of service (each level of service is an example of the service components referenced above.)
- NSP1 decides to only offer 3 of these levels of service, by creating a service definition that is derived from the one created by the WSP1 (i.e. inherits its attributes), but then marks two of the service levels inactive.
- When Joe and Fred run the service, the Customer Contact Agent (CCA) would present the 3 options, which they would select and run with. (As a reminder, Joe and Fred are one of thousands running this same service with NSP1 and one of tens of thousands overall within WSP1's network.)
- Now if the marketing group at NSP1 should decide it really should sell the other two levels of service. Without the inheritance approach, the marketing group would have to change the service definition for several thousand services so that they would then be able to use the other two service levels. With inheritance, a single service template (that is foundation for all instances of that service in RSP1) is changed and all their customers switch over immediately!
- WSP1 could likewise decide that upon further analysis by their application experts group, the third service level doesn't need quite as stringent a latency criteria, and could decide to modify the template for that 3rd service (instead of modifying tens of thousands of them) and it will take affect for all retail service providers and, in turn, for all of their customers!

1.4.4.2. Service Definition Continued

Referring back to Figure 26 , three key aspects are important.

- The blocks marked 430 indicate how the actual service definition 431 is comprised of the series of template definitions. The core definition is created by the WSP, the RSP then refines, then the Enterprise network manager defines for the enterprise as a whole.
- The end user view of a service definition in 431 is extremely simple: application is Netmeeting; User list (433) is Tom, Dick, etc.; and because of the template definition the user via the Customer Contact Agent (CCA) will direct the user to select a quality of service as low/med/high and a security level as on/off.
- The combination of user definitions, template definitions, and run time selection then create the overall information the NRAS needs to initiate this service. The template definitions that expand the user's selections into finite settings with which to configure the network equipment are shown in block 434-436. For instance, the user selected option of 'High' on the QoS would then be matched to an application template for Netmeeting to describe in generic terms what high QoS means for this application. This info would then be transformed in 435 into network specific terms (e.g. ATM would use cell delay and sustained cell rate.) The final transform into device specific terms is indicated by block 436.

Service Definition with Inheritance

NRAS implements a form of multiple inheritance in which objects with dissimilar properties that have completely independent inheritance histories are later joined to form an aggregate object which obtains it's behavior from diverse genealogy.

Figure 26 is an example that has been simplified to show such inheritance from two different collections of inherited attributes that come into the SB service definition system (SDS). First, there is the various service objects represented by a hierarchy where objects manipulated by role aware (WSP, RSP, etc.) tools at each level define service options and constraints for the next level down. Second, there is the inheritance of QoS definition objects from generic descriptions to network equipment specific configurations(e.g. 'high QoS for Netmeeting' to vendor specific settings for the network equipment itself.). The nature of a particular service definition instance is a combination of the service generic behavior inherited from the service objects and the QoS definition inherited from the Service QoS object.

Figure 28 shows an example of how this is constructed. Block 570 represents the top level definition that a WSP would create. (NOTE: even though it shows as one piece, it is still composed of multiple pieces, with different specialists or experts involved, as described above.) We will focus on the QoS aspect of the service definition for sake of our example. For example, block 570 might define a range of 7 possible QoS values usable in this service. Block 571 by the RSP might decide that only 5 are desirable for their customer profile. In block 572, the Enterprise network manager could in turn say that only 3 are authorized for his users. Block 573 represents an actual service definition instance (the ones above are considered templates), for instance the Netmeeting group with Bob, et al above.

The second aspect is to get actual settings associated with the QoS references above. Block 576 represents some core QoS definitions (typically setup by the infrastructure owner, in our example the WSP.) The set of seven definitions references above defined by the WSP definition 570 would be represented by a series of block similar to 577. When the actual Service Instance 573 is created, the three QoS choices are populated with actual values via block 580. (The generic definition 577 is converted into actual network settings via the mapping rules 578. These mapping rules are provided by an 'expert' on network topology and equipment such as a wholesale provider network engineer).

Once all of these references are resolved, the complete setting goes thru block 575 to convert into an XML based service definition 581. This is the definition used throughout the system. At this point the XML describes an object with behavior inherited from the combination of the inheritance tree for the QoS definition and the tree for the service definition.

As noted above, the key value add proposition for constructing service definitions is this way is the ease of definition and maintenance. Any component of the composite behavior can change without impacting the definitions in any of the other branches of the multiple inheritance tree. A WSP can change the underlying template to affect the operation of all services in the his network domain based on that template.

XML/DTD

The implementation of the NRAS uses XML to represent the individual components of the Service Definition, as well as the completed Service

Definitions. DTDs will define the exact XML structure for each of these components.

In this disclosure, there is shown and described only the preferred embodiment of the invention, but, as aforementioned, it is to be understood that the invention is capable of use in various other combinations and environments and is capable of changes or modifications within the scope of the inventive concept as expressed herein.

What is Claimed Is:

1. A network resource administration server comprising a translator for automatically translating user service requests into at least one form understandable by one of a specialist or network equipment.

2. The network resource administration server of claim 1 in which the different form is one understandable by one of a local enterprise network manager, an enterprise network manager, a retail service provider customer service representative, a wholesale service provider customer service representative and a network engineer.

3. The network resource administration server of claim 1 in which the network resource administration server control network elements at least one of a near side edge, far side edge and core of a network.

4. The network resource administration server of claim 3 in which the network resource administration server allocates resources in networks of different types.

5. The network resource administration server of claim 4 in which networks of different types include networks selected from the group comprising Ethernet, ATM, signalled, unsignalled, , diffserve, Frame Relay and Multi-Protocol Label Switching.

6. The network resource administration server of claim 1 connected to control at least one network.

7. The network resource administration server of claim 6 in which the network resource administration server is connected to parts of said at least one network through at least one concentrator.

8. The network resource administration server of claim 1 having a plurality of managers, one for controlling each of different parts of at least one network.

9. The network resource administration server of claim 1 connected to a second network resource administration server for administering respective networks or portions of a network.

10. The network resource administration server of claim 1 connected to a network for provisioning services to be provided by said network.

11. The network resource administration server of claim 1 for providing information about how to provision services to be provided by said network.

12. The network resource administration server of claim 1 for in part providing information about how to provision services to be provided by said network and in part provisioning said services for said network.

13. The network resource administration server of claim 1 configured to receive information about application startup at a point in the network and to automatically provide the network resources to support the network needs of that application.

14. The network resource administration server of claim 13, in which the network resource administration server monitors at least one communications stream for said application.

15. The network resource administration server of claim 14 in which network services are established and torn down based on information contained in said at least one communications stream.

16. A computing device, connected to a network, configured to detect start up of an application and to provide information to a component of said network about the communication needs of that application.

17. The computing device of claim 16, in which at least one communication stream sent to said network is marked with information about the communication needs of that application.

18. A computing device, connected to a network, configured to detect start up of an application and to provide information to a component of said network about the communication needs of that application using a browser process.

19. The computing device of claim 18, in which the browser process is contained in a plug in module for a browser.

20. The computing device of claim 19 in which the plug in module is configured to interact with a plurality of different quality of service capable plug in modules.

21. A computing device, connected to a network, configured to provide information to a user about quality of service options available to the user during a network session.

22. The computing device of claim 21 in which the information is provided in a user dialog window on a user's display.

23. The computing device of claim 21 in which the information is provided by a user workstation process.

24. The computing device of claim 21 in which the information is provided by a network interface device.

25. The computing device of claim 21 in which the information is provided by a network resource administration server.

26. The computing device of claim 21 in which the information is provided by one of a JAVA applet or a process generating Hypertext Mark Up Language.

27. The computing device of claim 21 in which the information is provided by an X Windows client interface.

28. A computing device, connected to a network, configured to detect start up of an application and to provide information to a network resource administration server about the communication needs of that application.

29. The computing device of claim 28 in which the computing device sends duplicated copies of a communications flow to both the network and to the network resource administration server.

30. The computing device of claim 28 in which the computing device routes at least one communication flow to the network resource administration which forwards the communication flow to the network.

31. A network resource administration server configured to receive requests for bandwidth allocations to be implemented at a future time.

32. A network resource administration server of claim 31 configured to maintain separate pools of bandwidth capability which can be allocated statically to requests for service and which can be allocated dynamically to requests for service, respectively.

33. A network resource administration server configured to provide billing information about resource consumption by application level processes.

34. The network resource administration server of claim 33, configured to provide a call detail records tracking resource consumption by flow groups based on time or resource usage.

35. The network resource administration server of claim 34, configured to provide call detail records to multiple destinations.

36. The network resource administration server of claim 35, in which the multiple destinations include users at least one of enterprise, retail service provider and wholesale service provider levels of a communication services distribution chain.

37. A network resource administration server configured to provide information about compliance with service level agreements during network use by application level processes.

38. The network resource administration server of claim 37, configured to provide a call detail records tracking service level agreement compliance by flow groups associated with an application.

39. The network resource administration server of claim 38, configured to provide call detail records to multiple destinations.

40. The network resource administration server of claim 39, in which the multiple destinations include users at least one of enterprise, retail service provider and wholesale service provider levels of a communication services distribution chain.

41. A network control configured to represent service requests in terms of logical rather than physical elements.

42. The network control of claim 41 in which the network control uses atomic virtual service elements and atomic virtual network elements to represent service requests in terms of logical elements.

43. The network control of claim 42 in which atomic virtual service elements and atomic virtual network elements are combined to create meta network elements for use in controlling network provisioning.

44. The network control of claim 43 configured to provision network services using said meta network elements.

45. The network control of claim 44 in which the same meta network elements are used to provision network services for network elements from different manufacturers.

46. The network control of claim 45 in which meta network elements control settings of physical network elements from different manufacturers using device drivers for each manufacturer for a particular type of network element.

47. A network resource administration server comprising a rules based translator for automatically translating user service requests into at least one form understandable by one of a specialist or network equipment.

48. The network resource administration server of claim 47, in which rules are arranged heirarchically.

49. The network resource administration server of claim 48 in which respective rule sets are maintained for different levels of a communications services distribution hierarchy.

50. The network resource administration server of claim 48 in which rule sets are maintained for at least two of zone, enterprise, virtual private network and service.

51. The network resource administration server of claim 47 in which rules for implementing a service are provided to network elements only when a network element is actually needed to handle a service request.

52. The network resource administration server of claim 51 in which the rules for implementing a service are removed from a network element once the service request has been satisfied.

53. The network resource administration server of claim 47, in which the expansion of rules into lower level rules may be modified based on at least one of network load, reservations for a service, user selection of quality of service and financial status.

54. The network resource administration server of claim 47 in which rules needed for provisioning a particular service originate from a plurality of sources.

55. A method of provisioning networks, comprising the steps of:

a. receiving a request for service using terminology appropriate to a user;

b. receiving information from one or more specialists to be combined with information received from said user; and

c. automatically provisioning the network to provide a service requested by said user.

56. The method of claim 55 in which service definitions are stored using XML DTD.

57. A method of provisioning a network to provide service, comprising the step of providing rules for implementing a service to network elements to configure the network element just at the time needed to actually provide the service.

58. The method of claim 57 comprising the additional step of removing the rules from a network element when the service has been completed.

59. A method of simplifying rules administration in a policy based network, comprising the steps of:

a. defining a template rule for at least one level of a rules hierarchy, each level of the hierarchy corresponding to steps required for network provisioning, and

b. at least one rule at a particular level of the rules hierarchy inheriting attributes and properties from a rule at a higher level of the rules hierarchy.

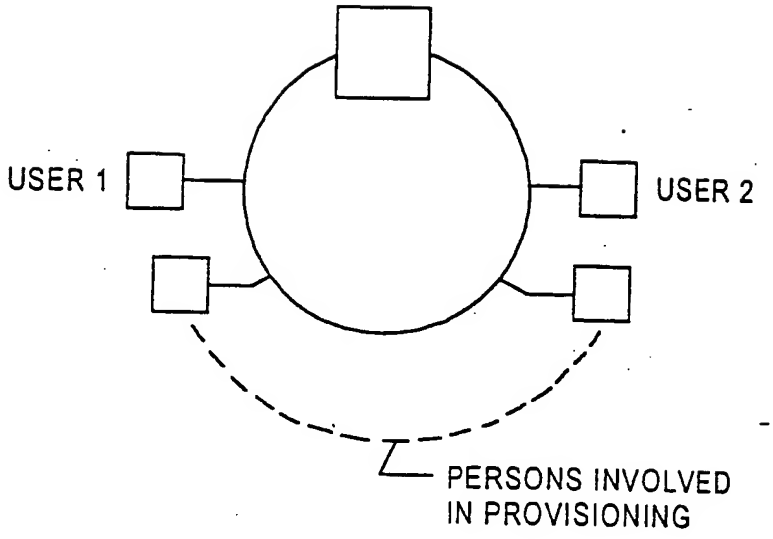


FIG. 1

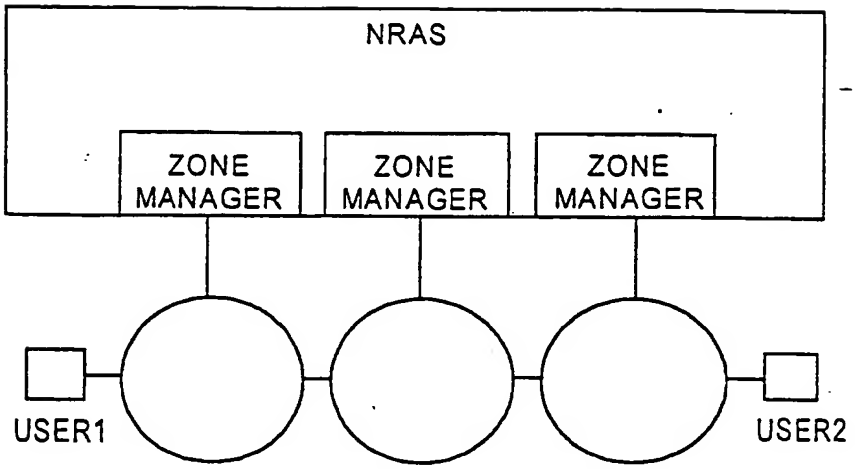


FIG. 2A

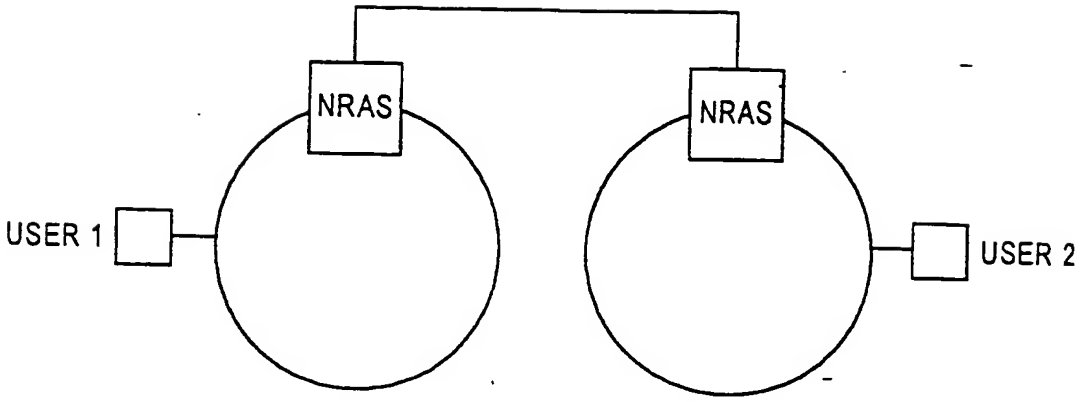


FIG. 2B

FIG 3

NRAS Functional components and positioning in network

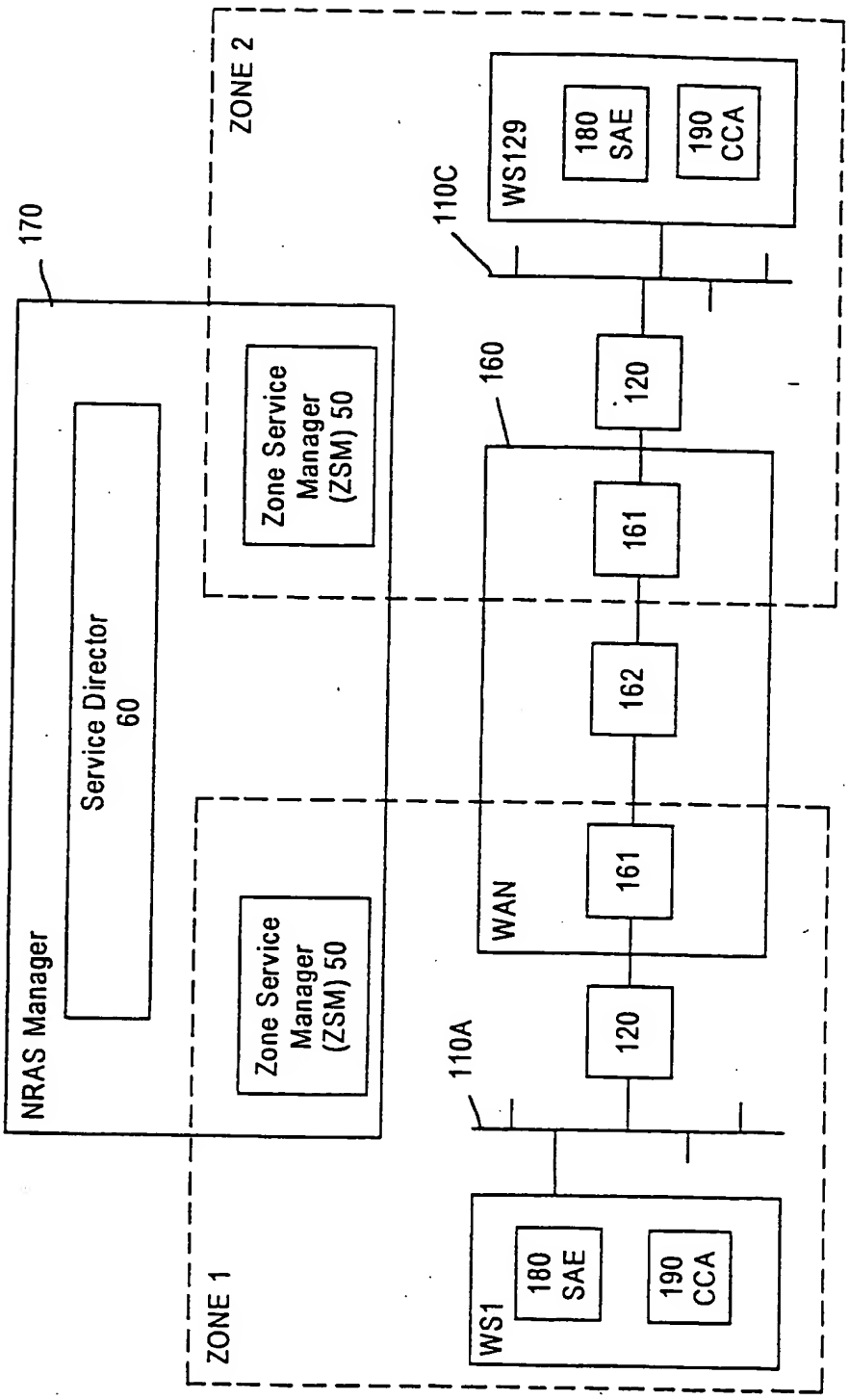


FIG. 4
Physical Topology

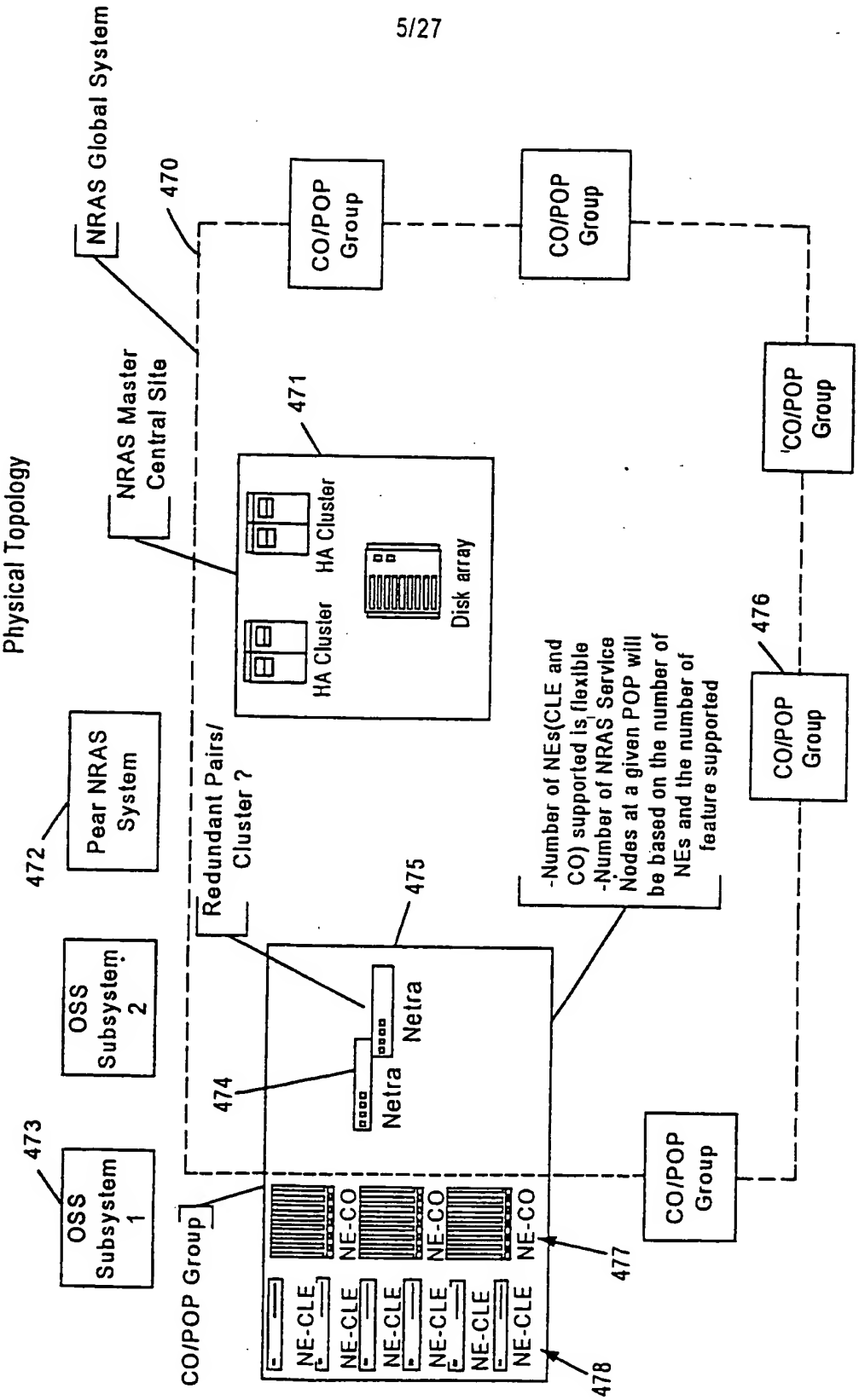
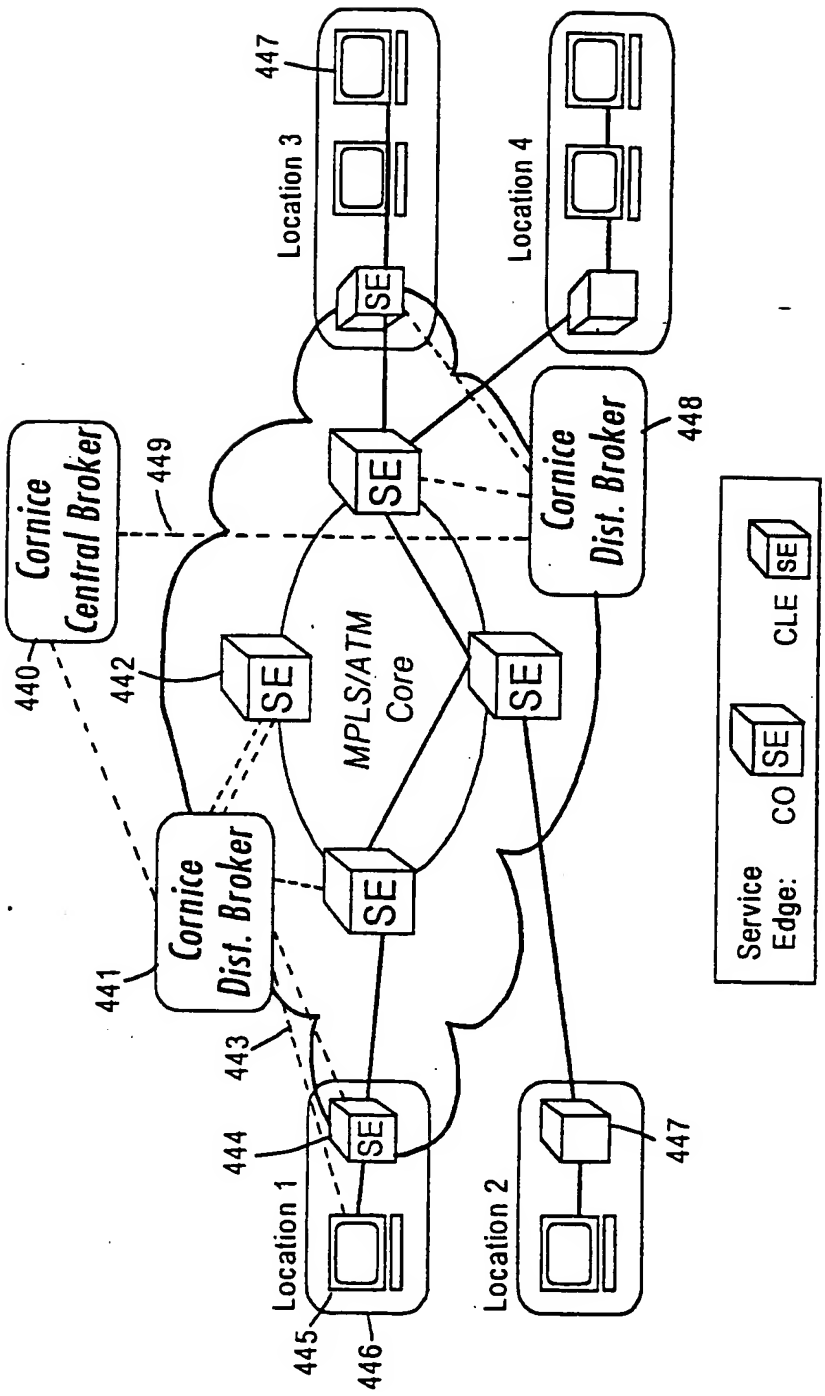


FIG. 5

Service Broker Position in Network



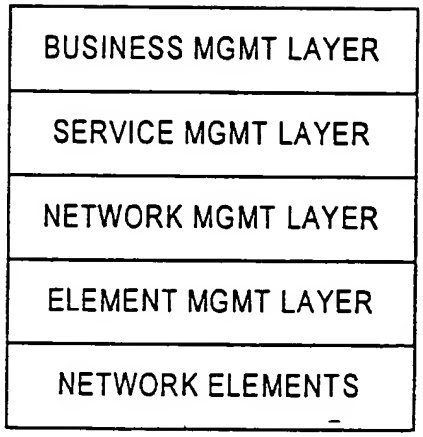

BUSINESS MGMT LAYER
SERVICE MGMT LAYER
NETWORK MGMT LAYER
ELEMENT MGMT LAYER
NETWORK ELEMENTS

FIG.6

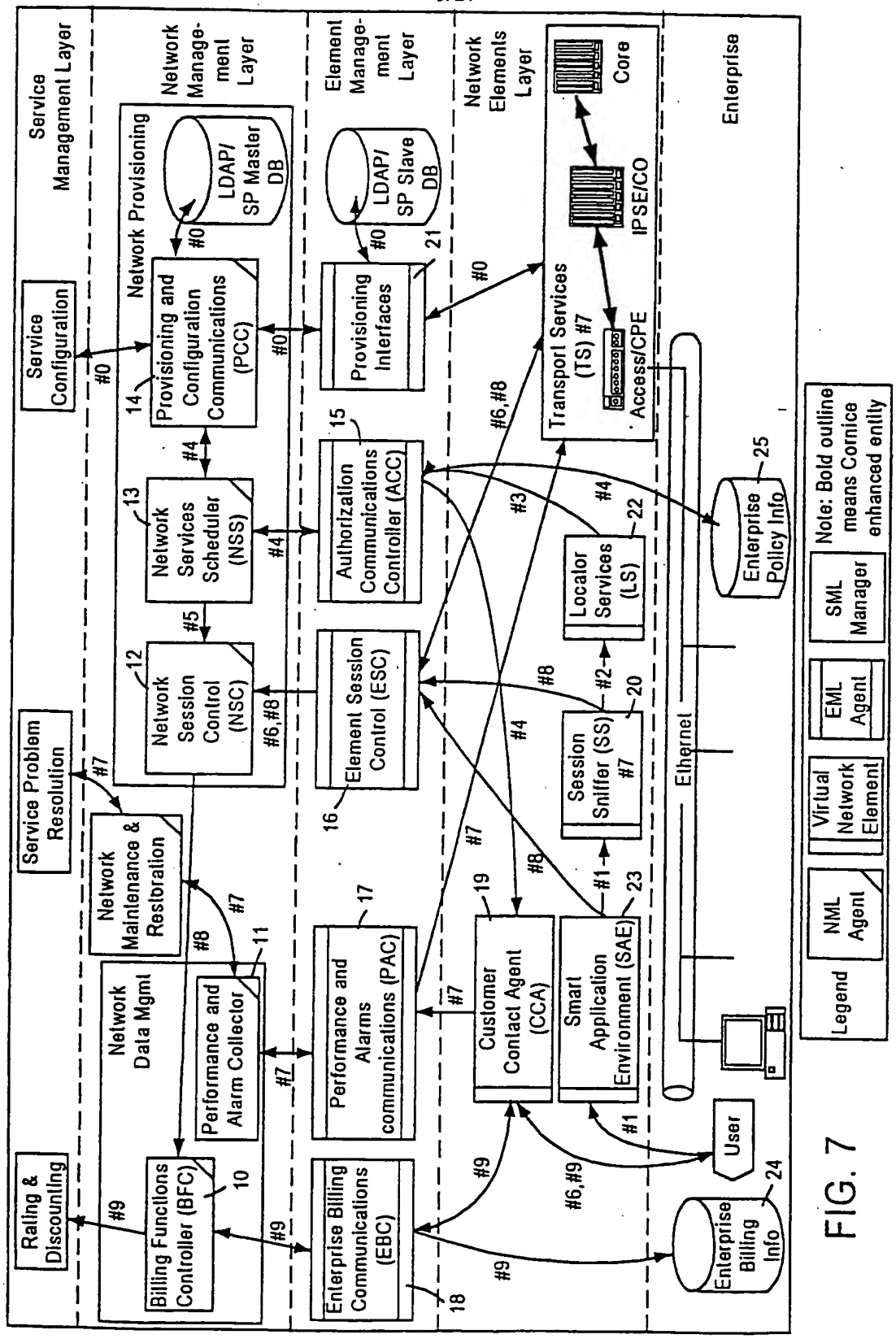

Service Management Layer
Network Management Layer
Element Management Layer
Network Elements Layer
Enterprise
Service Configuration
Service Problem Resolution
Rating & Discounting
Network Data Mgmt
Billing Functions Controller (BFC)
Network Maintenance & Restoration
Performance and Alarm Collector
Enterprise Billing Communications (EBC)
Performance and Alarms communications (PAC)
Network Provisioning
Provisioning and Configuration Communications (PCC)
Network Services Scheduler (NSS)
Network Session Control (NSC)
LDAP/ SP Master DB
LDAP/ SP Slave DB
Provisioning Interfaces
Authorization Communications Controller (ACC)
Element Session Control (ESC)
Customer Contact Agent (CCA)
Smart Application Environment (SAE)
Session Sniffer (SS) #7
Locator Services (LS)
Transport Services (TS) #7
Core
IPSE/CO
Access/CPE
Enterprise Policy Info
Enterprise Billing Info
Ethernet
User
Legend
NML Agent
Virtual Network Element
EML Agent
SML Manager
Note: Bold outline means Cornice enhanced entity

FIG. 7

FIG. 8
SAE- Embedded in Workstation

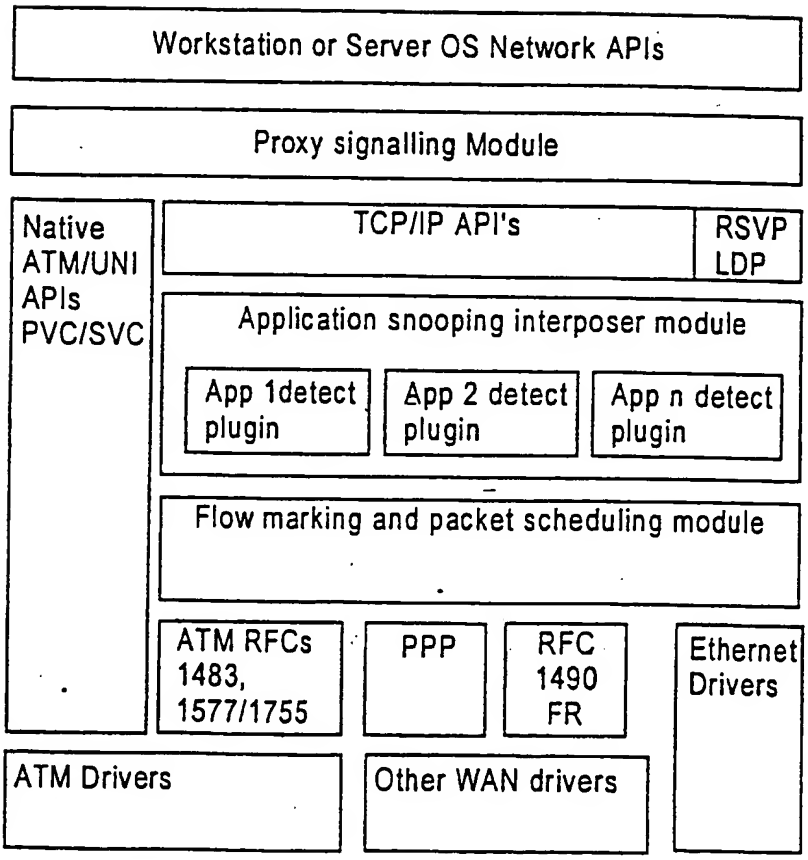


FIG. 9

CCA User to IP Address Mapping Flowchart

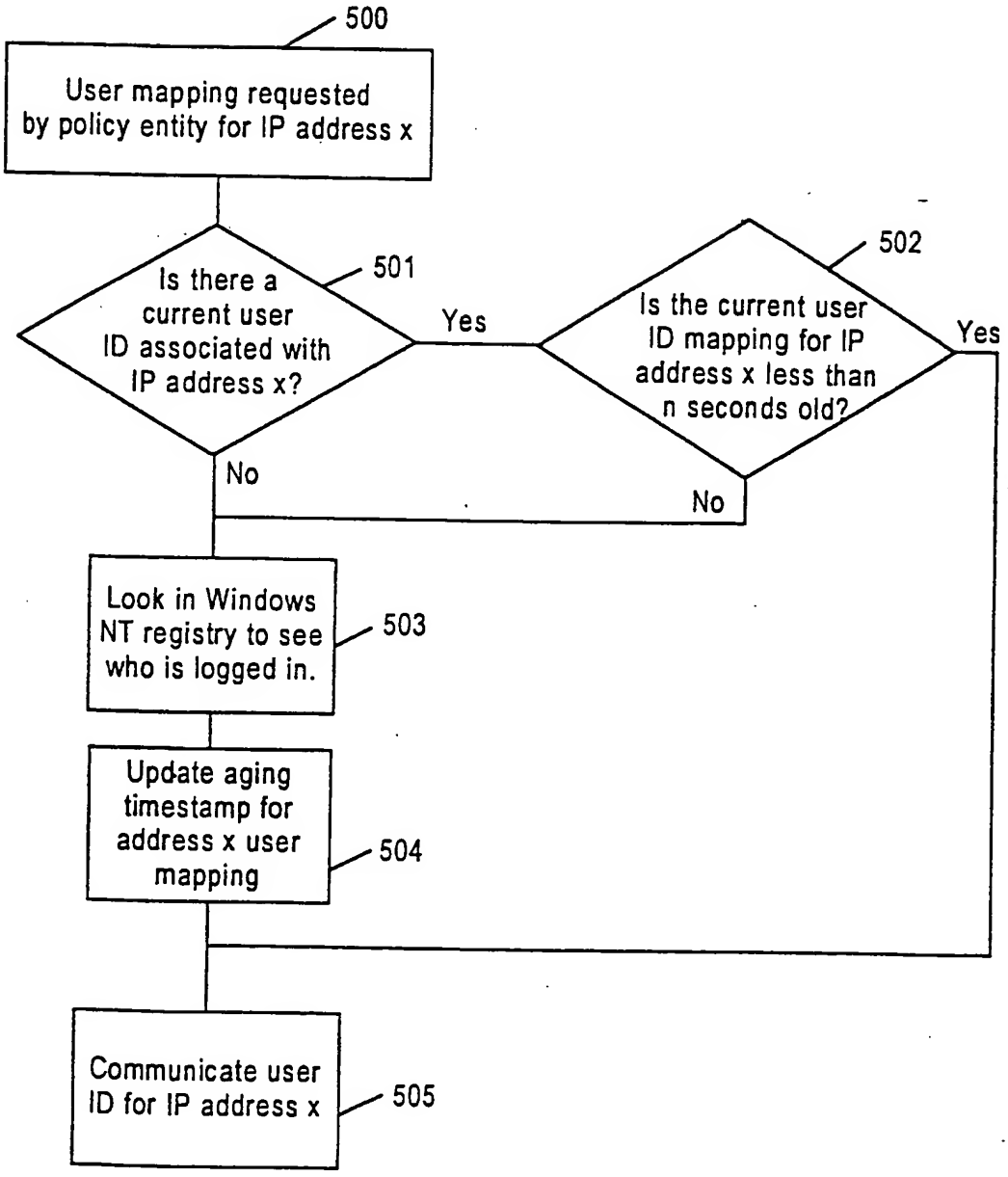


FIG 10

SAE Proxy Signalling Module Flowchart

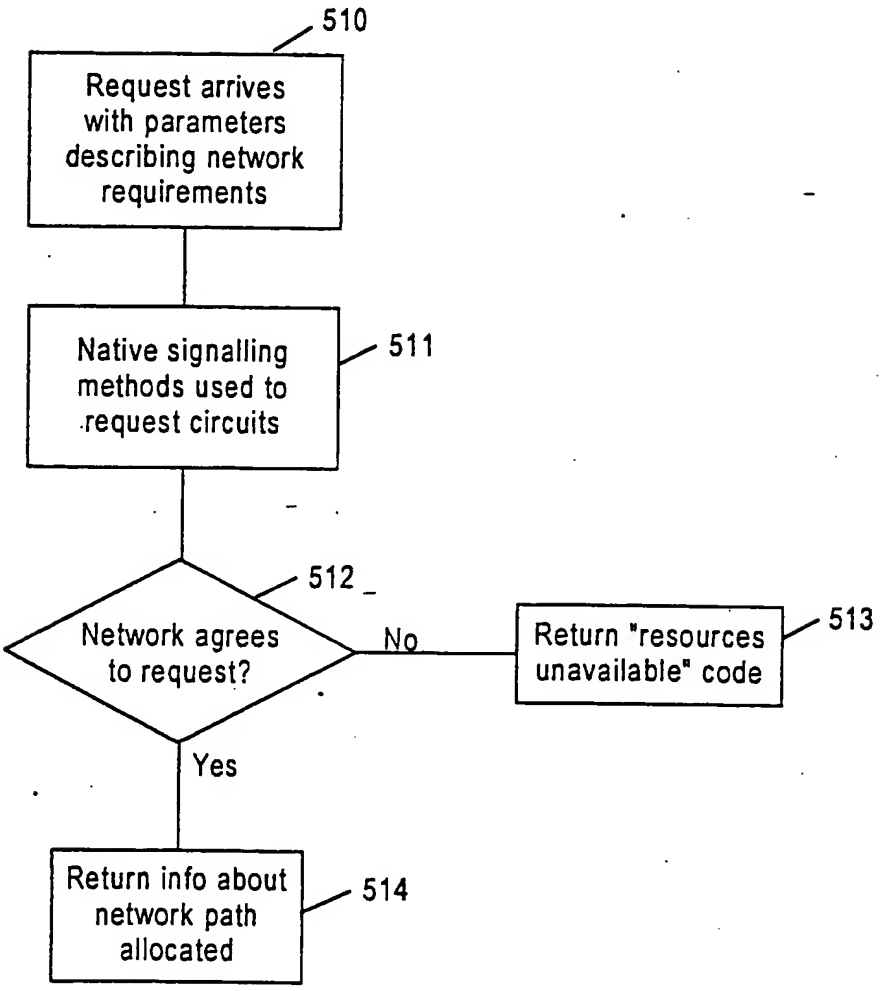


FIG. 11
SAE Marking and Scheduling Module Flowchart

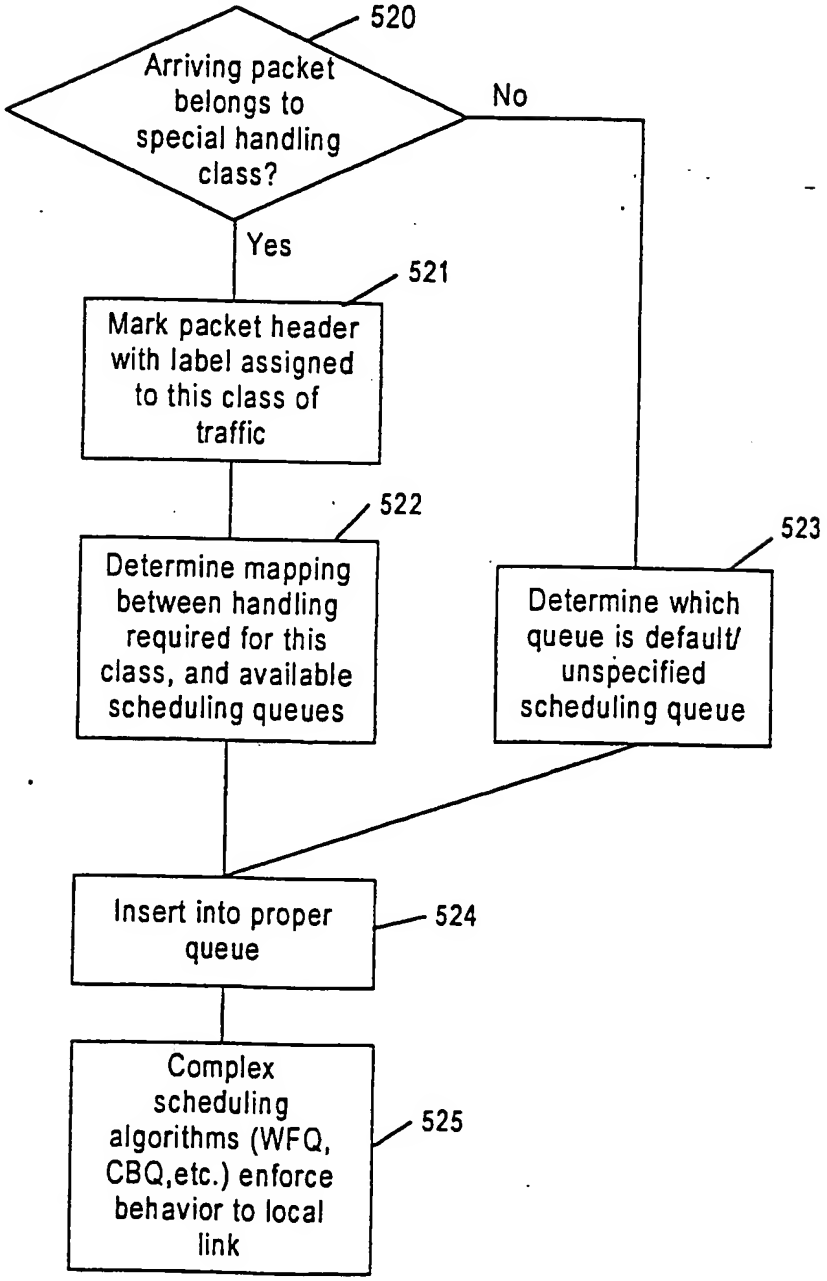


FIG. 12
Sniffer Block Diagram

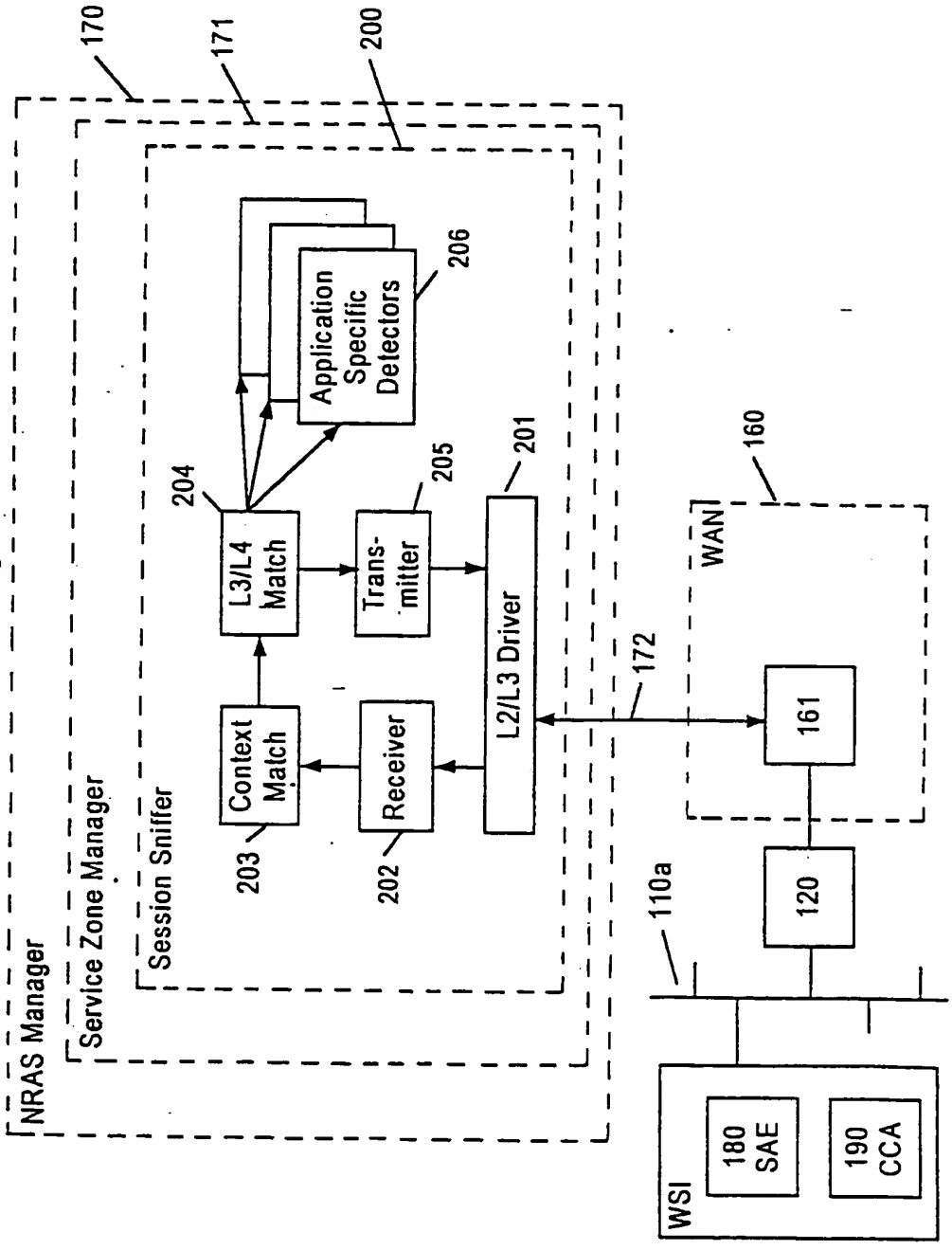


FIG. 13
Sniffer Flow Chart

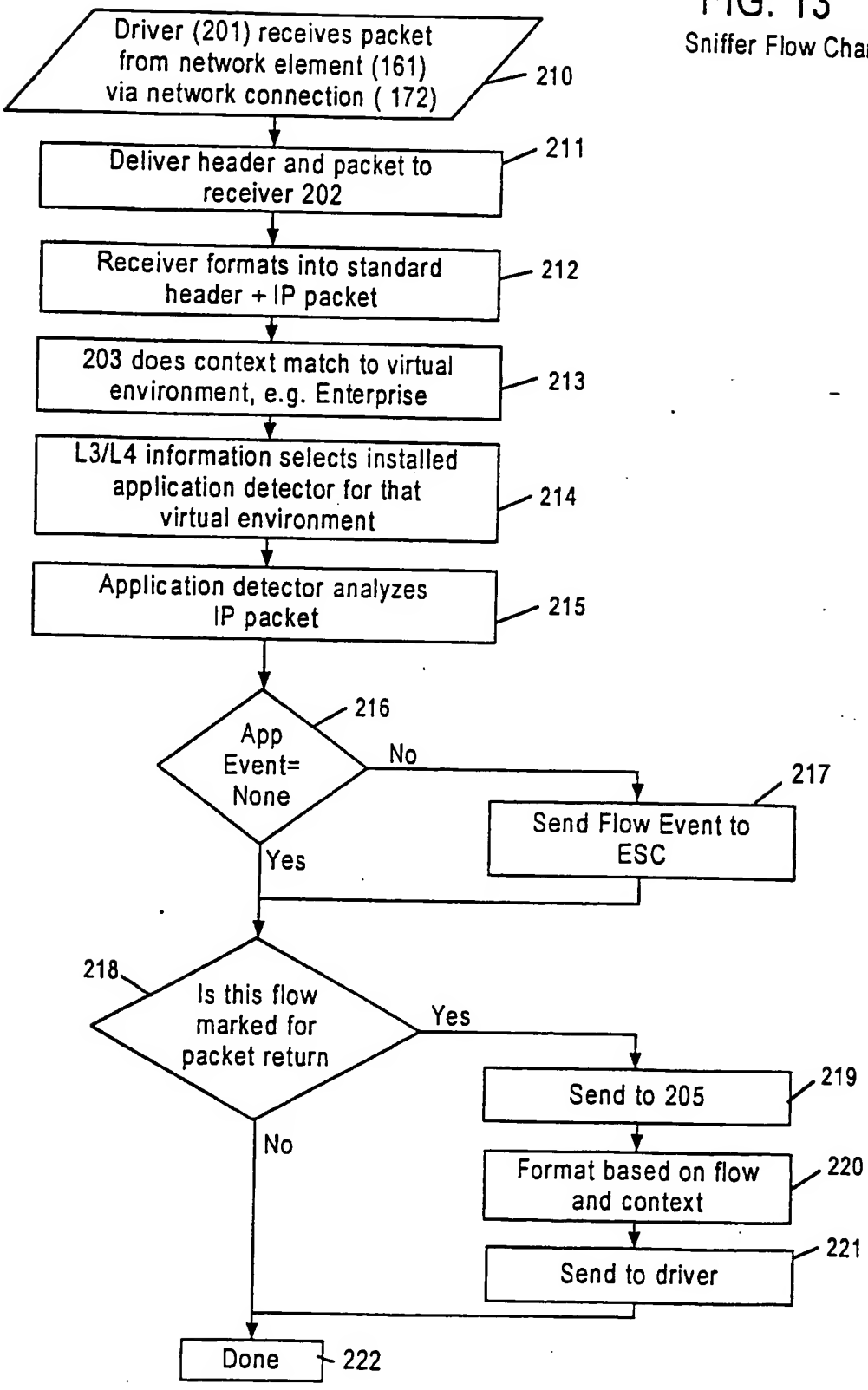


FIG. 14
Step 3 Flow Chart

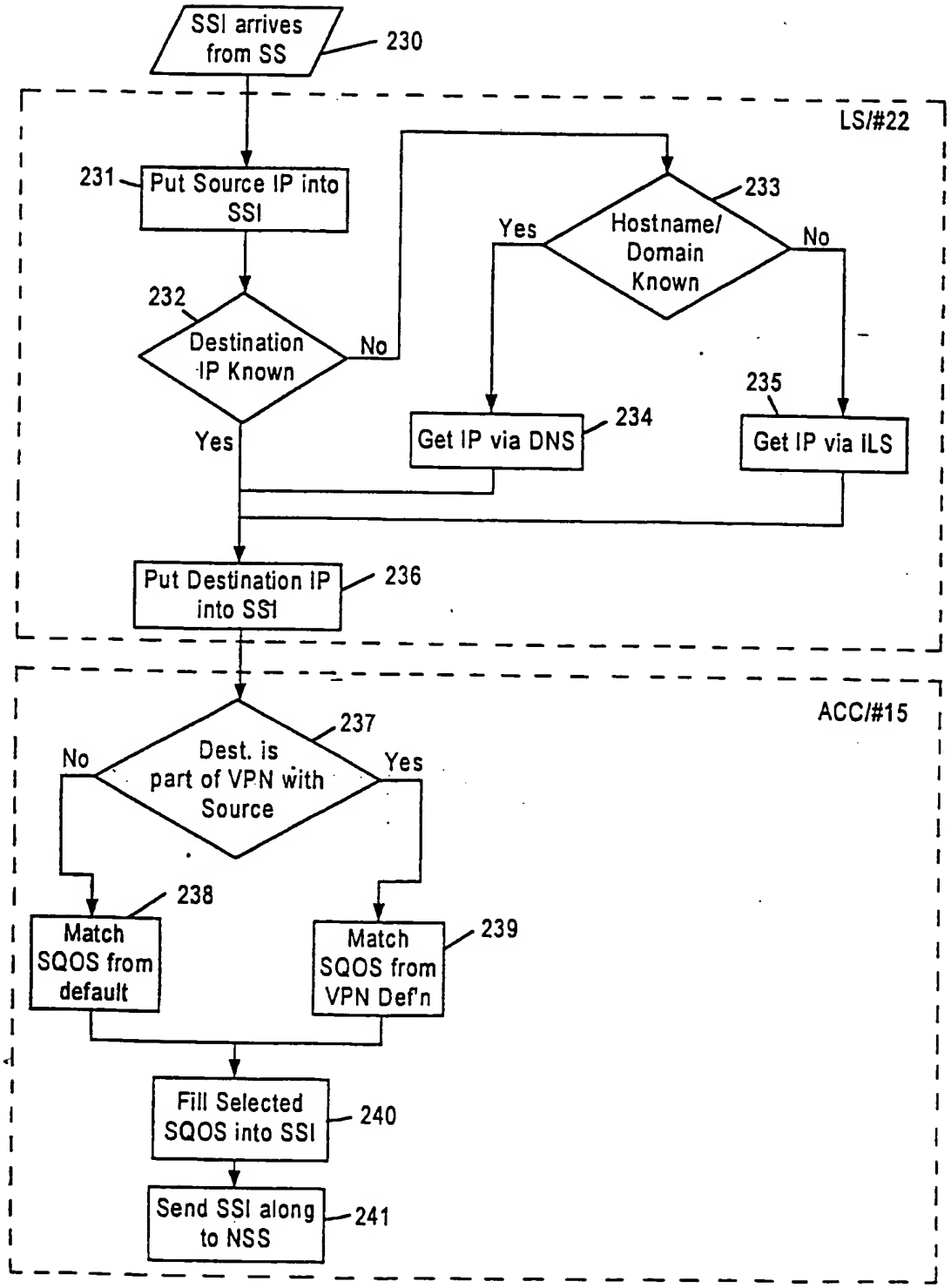


FIG. 15
Step 4 Flow Chart

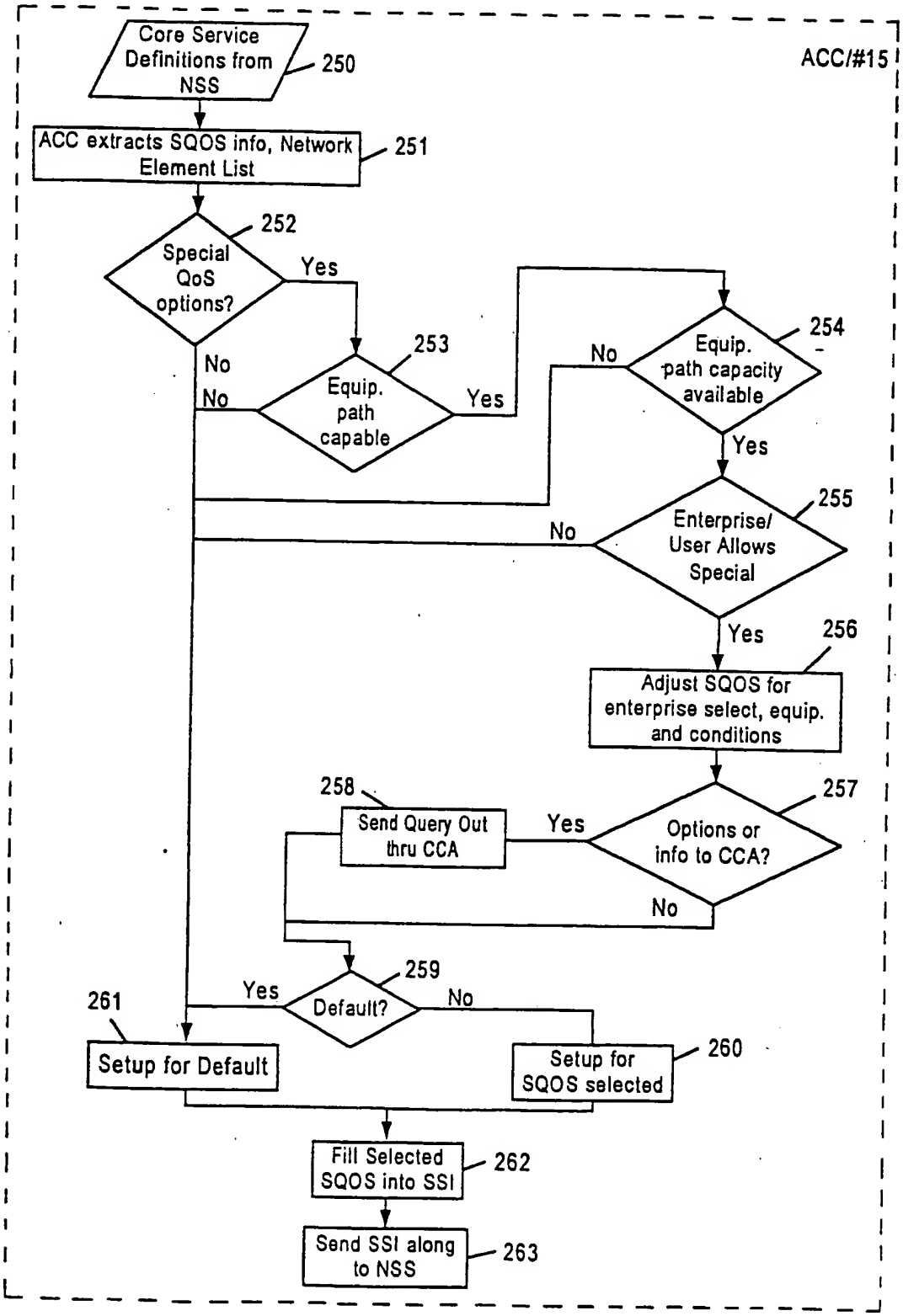


FIG. 17
NSS Flowchart

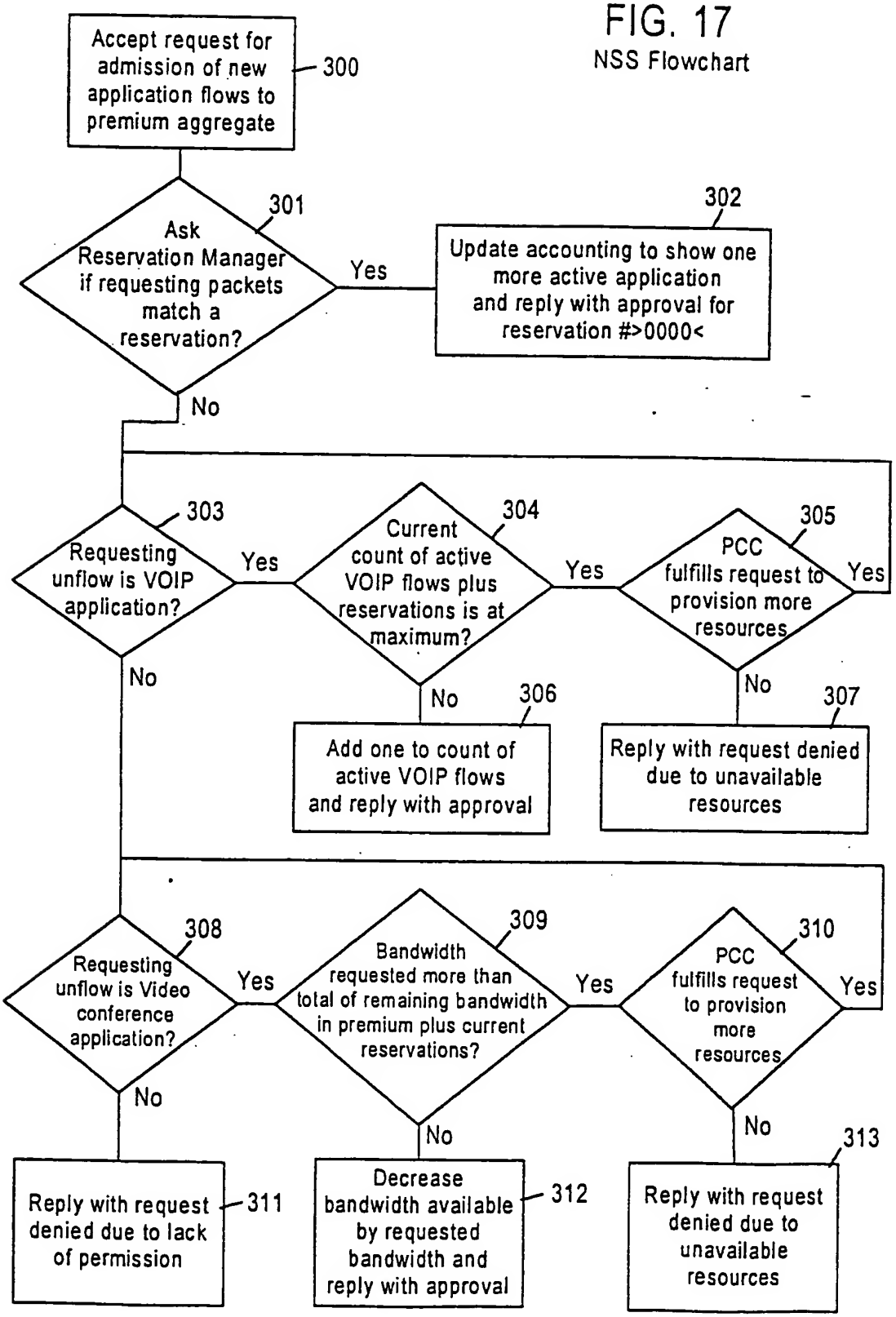


FIG. 18

Step 6 Flow Chart

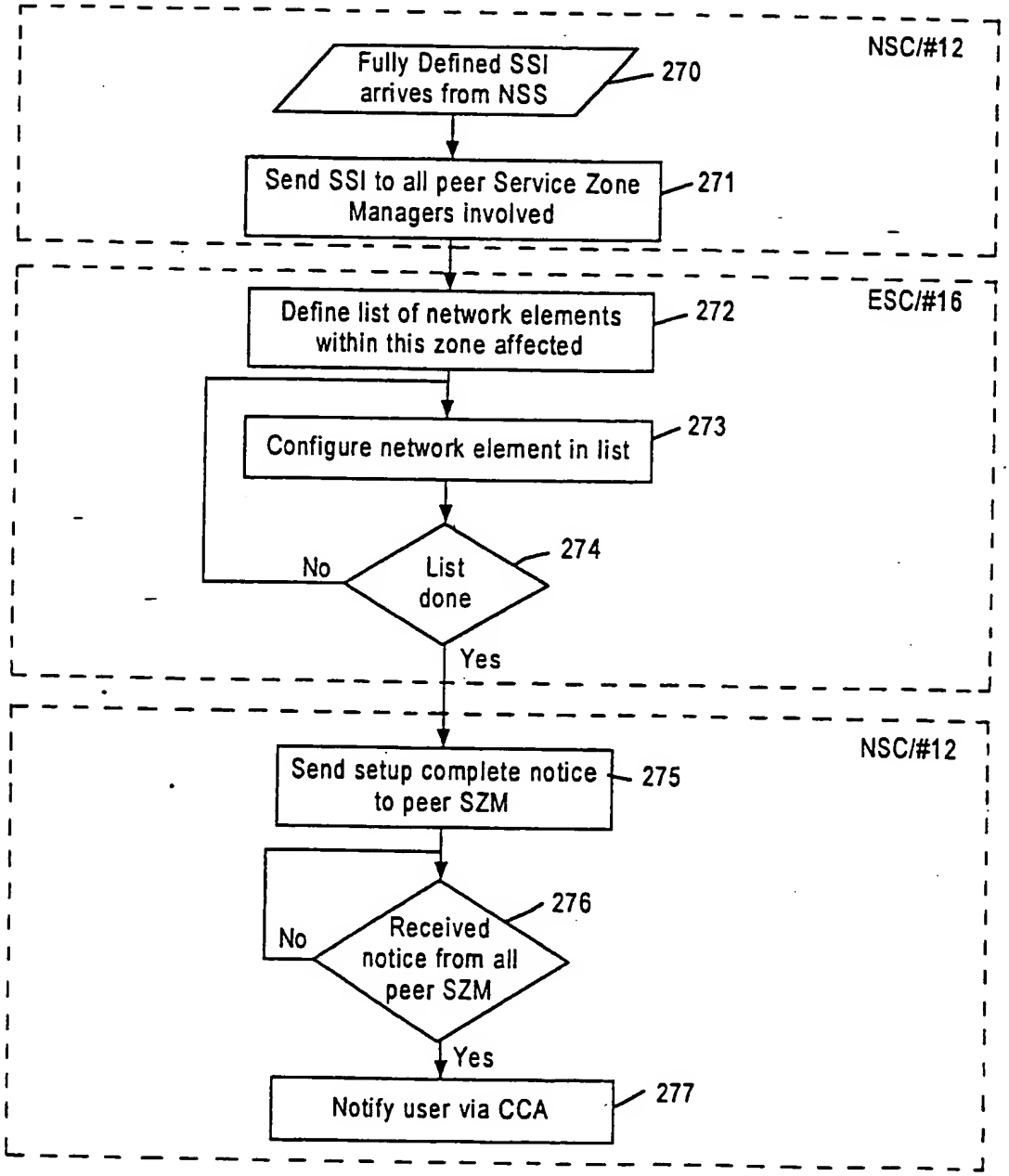


FIG. 19

meta Element Controller (MEC)

One per VPN, 'ideal' NE model (can do everthing the VPN needs for this Service

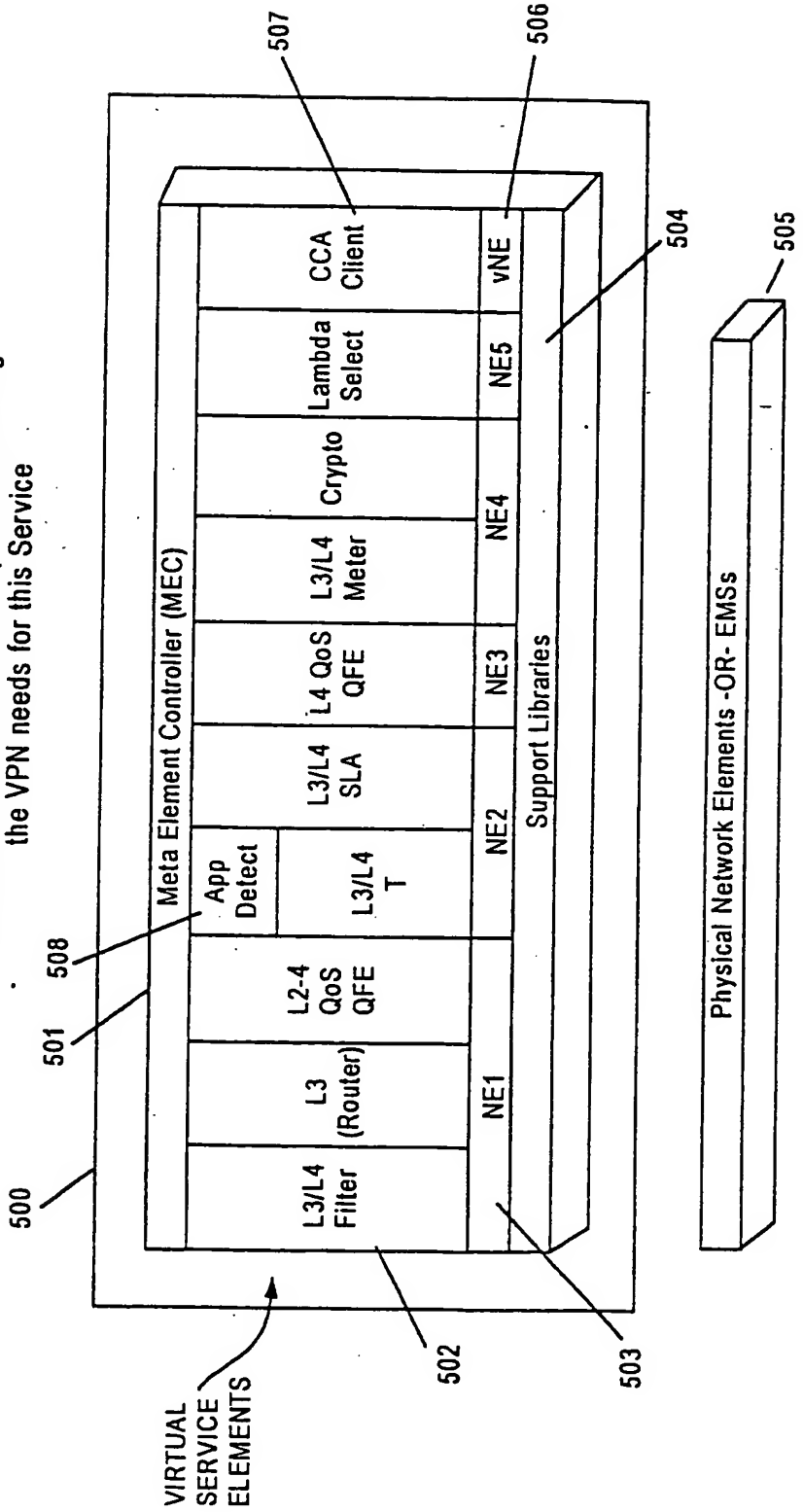


FIG. 20

Session Start Information

Item	Note
Source IP Address	Actual IP address is usually known
Destination IP Address/Name	Can either be actual IP address or URL User or...
Originating User/Enterprise	
Start Time	
Service QoS Options Array	List of possible options
Billing Options Array	List of possible options, includes: type, format, destination
User Notification Info	
CCA Options Array	List of options related to CCA display: '.gif' URL and display location,..

FIG. 21

Session End Information

Item	Note
Source IP Address	Actual IP address
Destination IP Address	Actual IP address
Originating User/Enterprise	
Destination Server/User/Enterprise	Could be server or user
Start Time	Per time zone of originating node
End Time	
Service QoS Used	
Application Information	
Per direction packets counts	Optional
Per direction byte counts	Optional
SLA results	Optional

FIG. 22
Step 6 Flow Chart

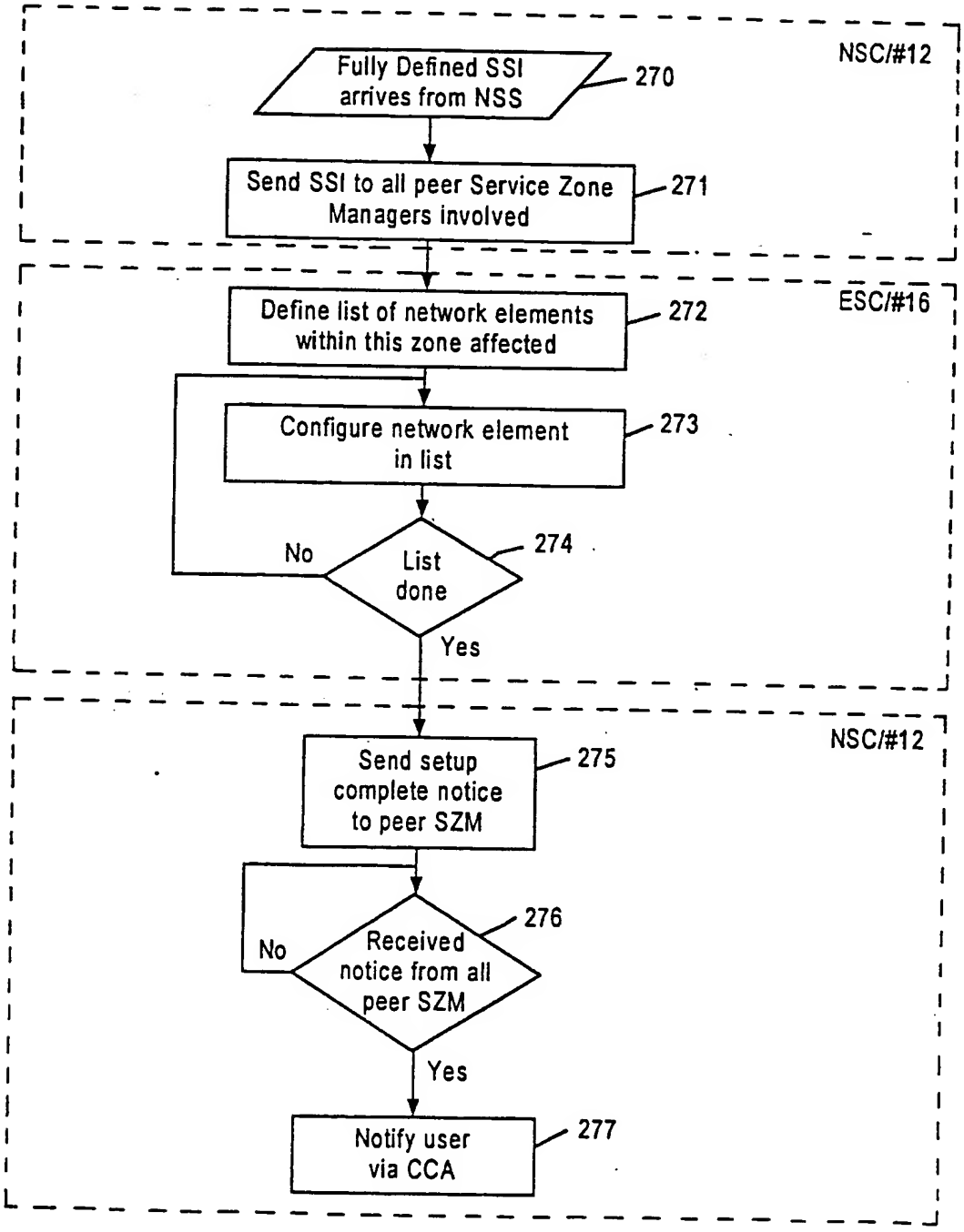
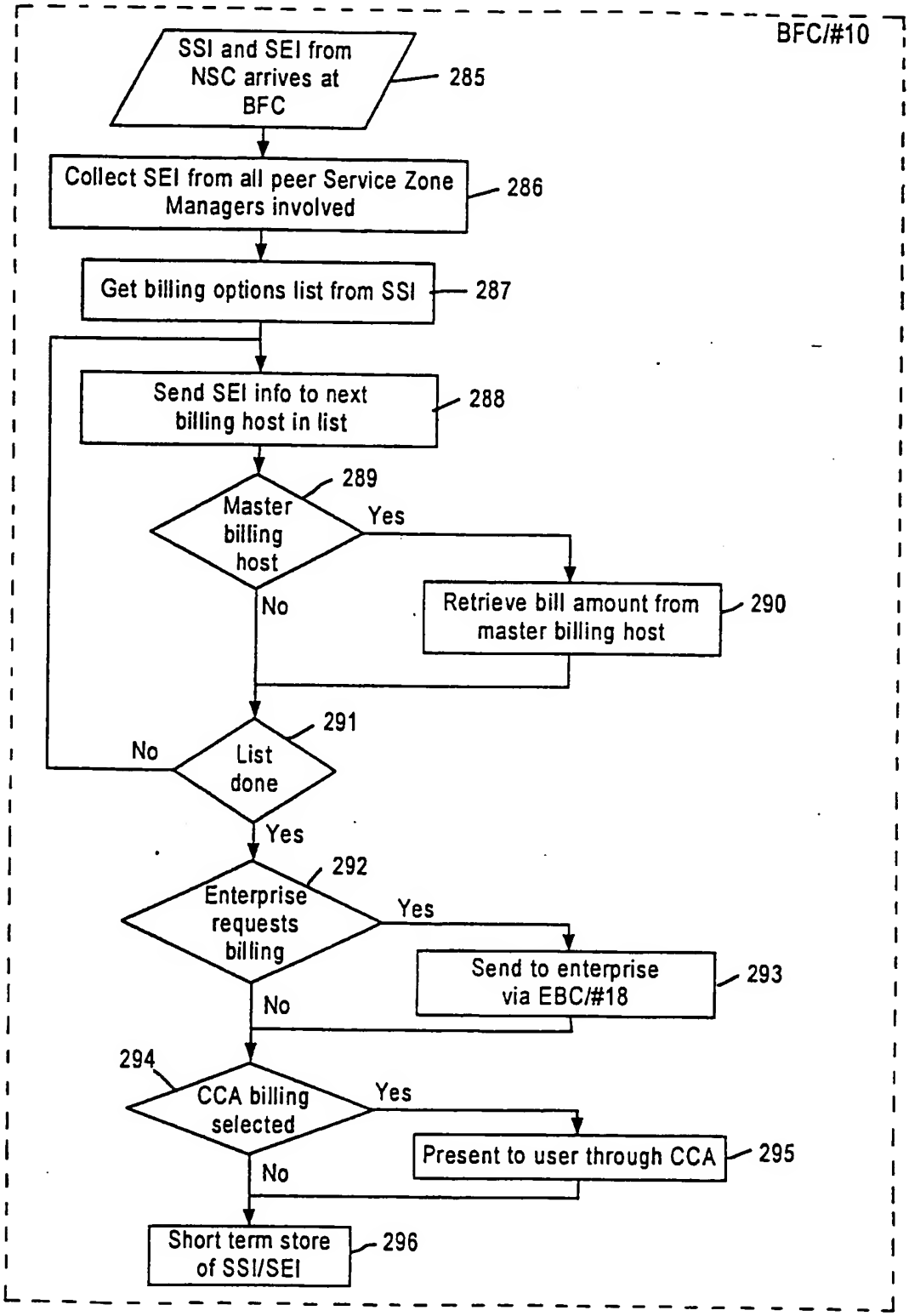


FIG. 23
Step 9 Flow Chart



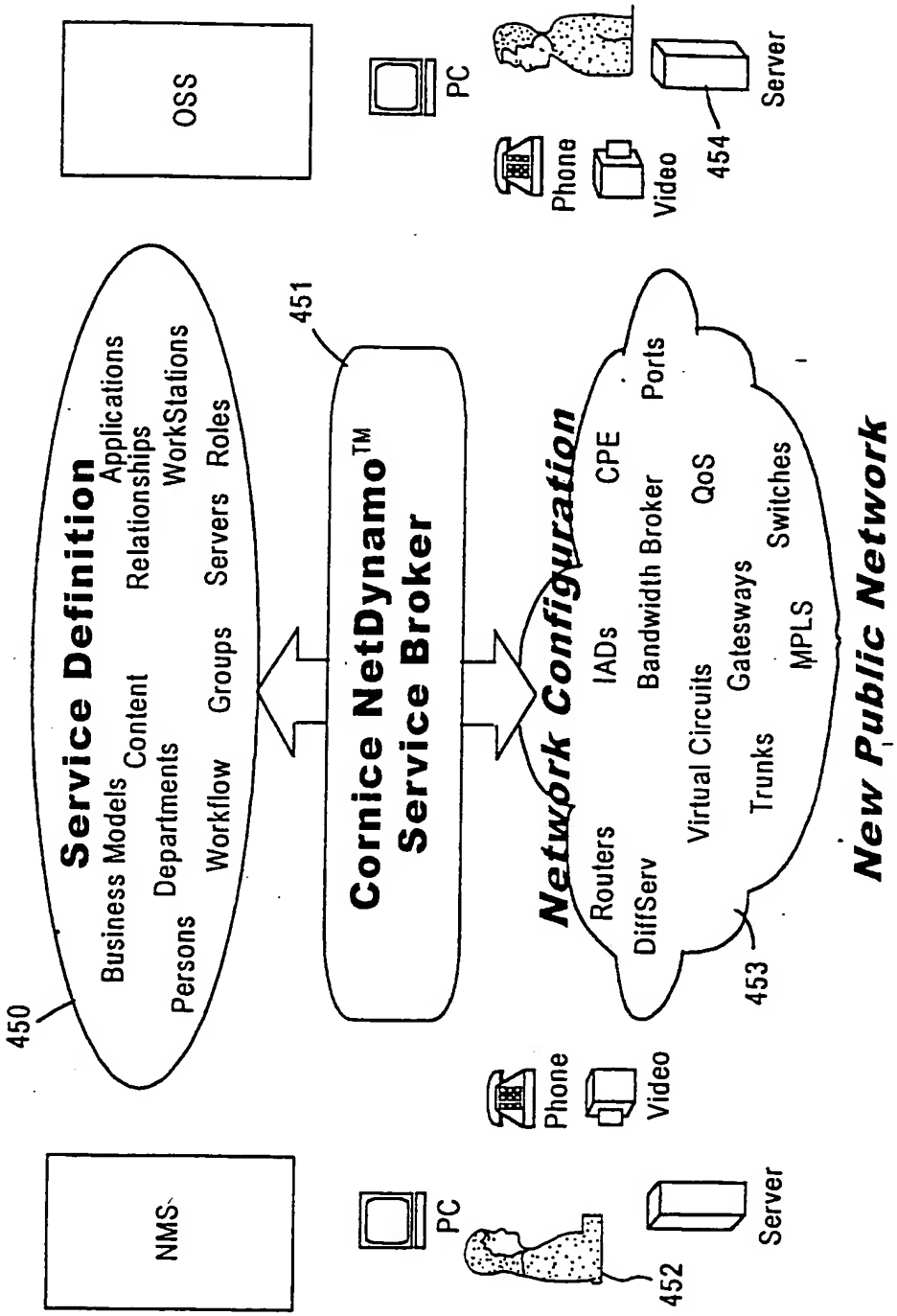


FIG. 24
Service Broker - Service Automation Layer in the Network

FIG. 25

Service Definition Construction

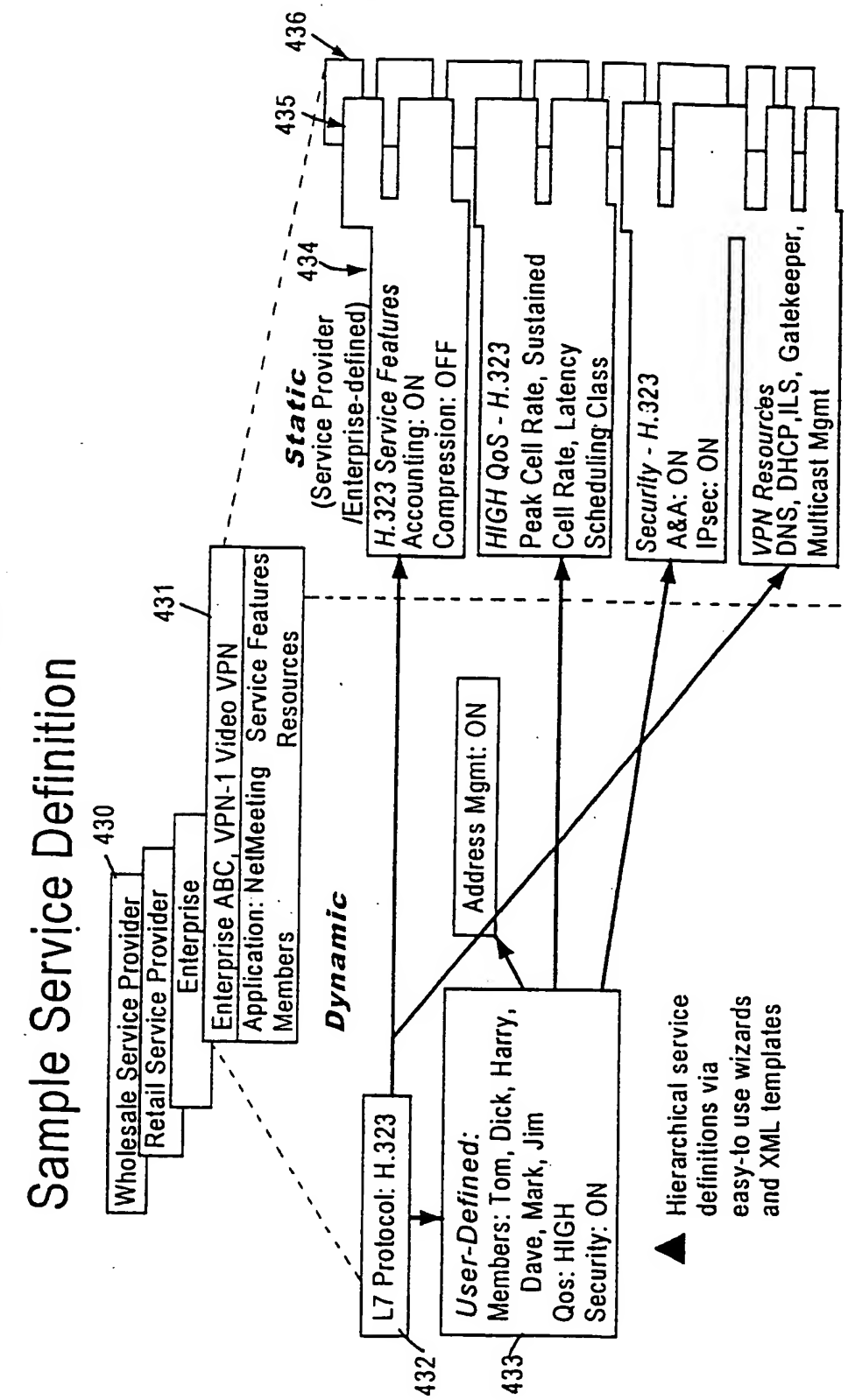


FIG. 26
Service Broker Overview

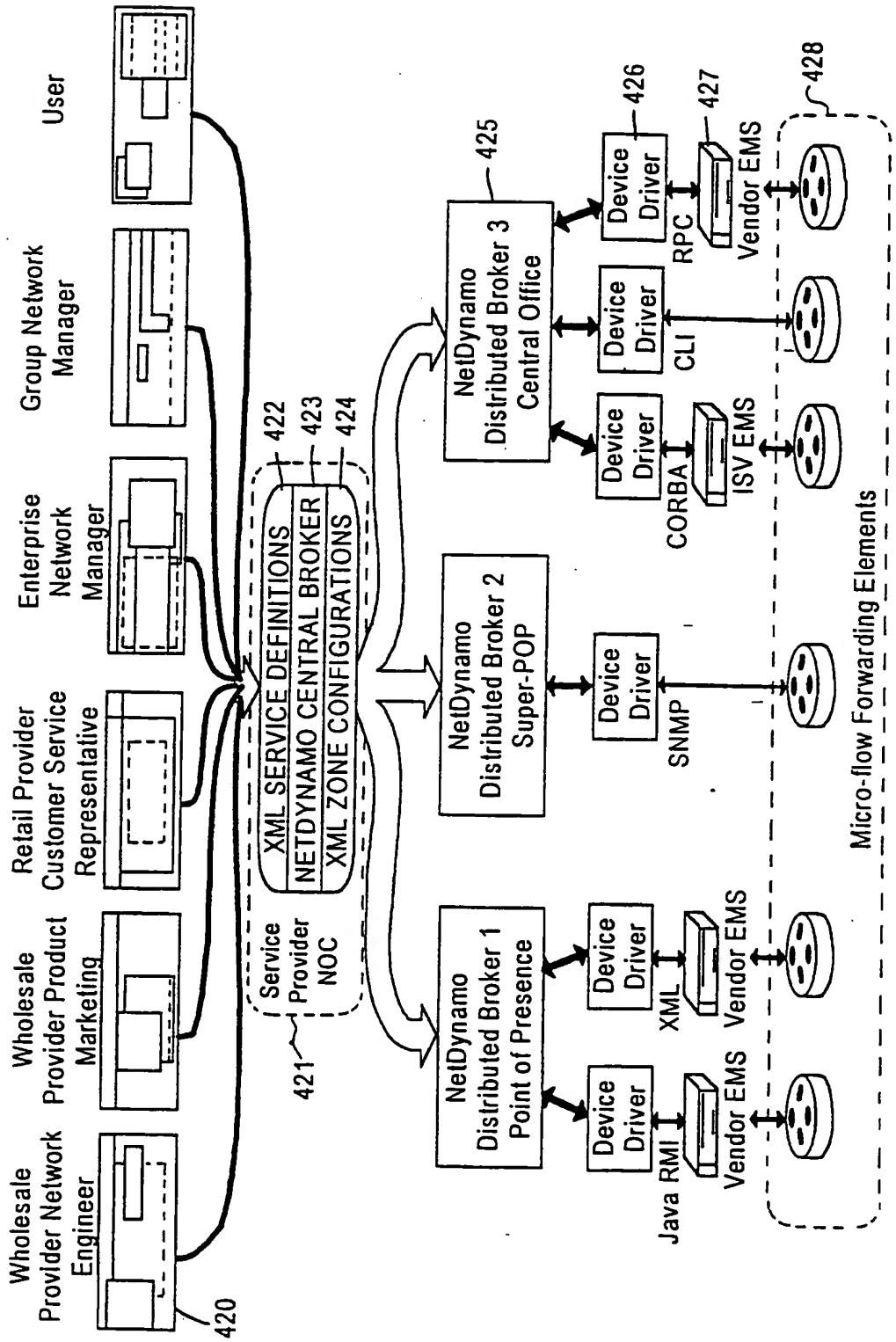
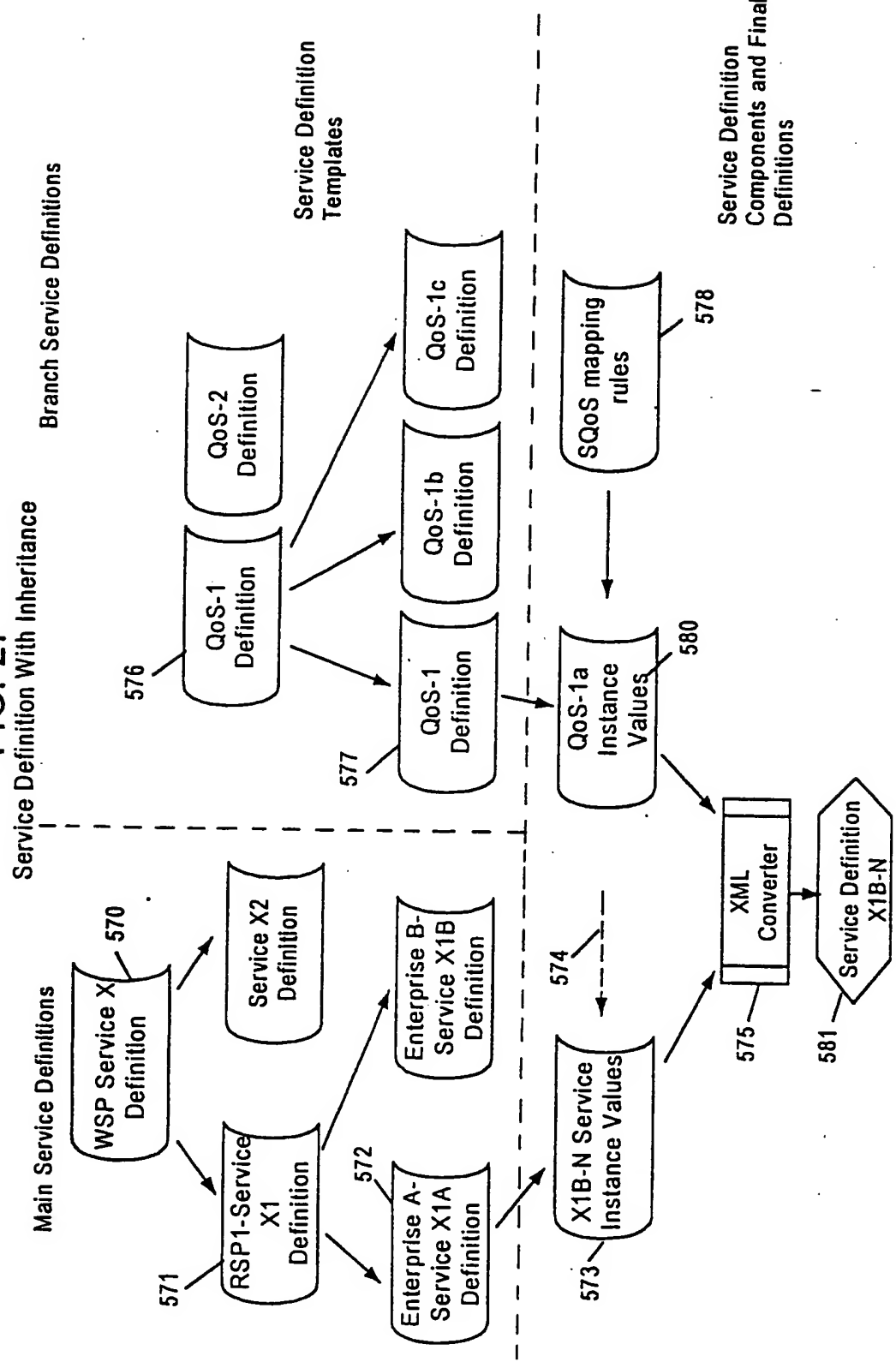


FIG. 27


Service Definition With Inheritance
Branch Service Definitions
Service Definition Templates
Service Definition Components and Final Definitions
Main Service Definitions
WSP Service X Definition
570
Service X2 Definition
RSP1-Service X1 Definition
571
Enterprise B-Service X1B Definition
Enterprise A-Service X1A Definition
572
574
X1B-N Service Instance Values
573
QoS-1 Definition
576
QoS-2 Definition
QoS-1c Definition
QoS-1b Definition
QoS-1 Definition
577
SQoS mapping rules
578
QoS-1a Instance Values
580
XML Converter
575
Service Definition X1B-N
581

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/07577

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 15/173, 15/16

US CL :709/223, 224, 225, 226, 249; 705/8; 370/352, 389

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/223, 224, 225, 226, 249; 705/8; 370/352, 389

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST, STN, DIALOG

search terms: network resource, allocate, monitor, packet, analysis, QOS, TOS, Bandwidth, translate or translation

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,748,892 A (RICHARDSON) 05 MAY 1998, FIG. 1, ABSTRACT	1-59
A	US 5,715,395 A (BRABSON ET AL.) 03 FEBRUARY 1998, COL. 9, LINES 26-64	1-59
Y	US 5,870,545 A (DAVIS et al.) 09 FEBRUARY 1999, FIGS. 2, 4, 6, COL. 4, LINES 29-67, COL. 5, LINES 1-67, COL. 6, LINES 1-67	1-59
A	US 5,838,921 A (SPEETER) 17 NOVEMBER 1998, ABSTRACT	1
A	US 5,884,037 A (ARAS et al.) 16 MARCH 1999, FIGS. 1, 2, AND 3, ABSTRACT	1-4

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance, the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
15 JUNE 2000	06 JUL 2000

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	AHMAD MATAR For Eugenia Logan
Facsimile No. (703) 305-3230	Telephone No. (703) 305-4731

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/07577

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,867,494 A (KRISHNASWAMY et al.) 02 FEBRUARY 1999, COL. 29, LINES 55-67, COL. 30, LINES 1-67, COL. 31, LINES 1-67, COL. 32, LINES 1-67 COL. 34, LINES 33-67, COL. 35, LINES 1-67	1-59